

All rights reserved. This document contains proprietary and confidential material, and is only for use by licensees of the SyncSort for z/OS proprietary software system.

P R O V E N
performance

SyncSort for z/OS

Installation Guide

Release 1.1

SI-4302-3
1.1c/RN-1

© Syncsort Incorporated, 2003

All rights reserved. This document contains proprietary and confidential material, and is only for use by the licensees of the SyncSort proprietary software system. This publication may not be reproduced in whole or in part, in any form, except with written permission from Syncsort Incorporated.

SyncSort and Visual SyncSort are trademarks of Syncsort Incorporated. No claim is made to the exclusive right to use *Visual* apart from the mark as shown. All other company and product names used herein may be the trademarks of their respective companies.

Table of Contents

Chapter 1.	Introduction	1.1
	Interactive Installation	1.1
	Non-Interactive Installation	1.1
	Structure of the Installation Guide.....	1.1
	Installation Considerations.....	1.2
	System Resident or Non-Resident Configuration	1.3
	Installing the BetterGener Facility	1.3
	PARASORT and Special Esoteric Unit Names	1.3
	Dynamic Storage Management (DSM)	1.4
	Global DSM Functions	1.5
	Local DSM	1.6
	Visual SyncSort for z/OS	1.6
	Notational Conventions	1.6
	Related Reading	1.7
Chapter 2.	SYNCINIT.....	2.1
	Installing SYNCINIT.....	2.1
	Invoking SYNCINIT	2.4
	Restoring Delivered Defaults	2.4
Chapter 3.	Installation (Non-SMP)	3.1
	Preliminary Tasks	3.1
	ELPA Storage Requirements.....	3.3
	DASD Space Requirements.....	3.3
	Non-SMP Installation Procedure	3.4
	Activating Global DSM	3.9
	The START Command.....	3.10

	Automatic Activation of Global DSM.	3.11
	Deactivating Global DSM.	3.11
	PARASORT.	3.12
	Activating BetterGener.	3.12
	Activating SS11MSG.	3.14
	SMFDSECT Location.	3.15
	Returning SYNCLIST Output and the Installation Tape.	3.15
	Maintenance.	3.16
Chapter 4.	Installation (SMP/E).	4.1
	Required Tape Files.	4.1
	Preliminary Tasks.	4.2
	SYSMODs.	4.5
	DDnames for Target and Distribution Libraries.	4.5
	ELPA Storage Requirements.	4.6
	DASD Space Requirements.	4.6
	SMP/E Installation Procedure.	4.7
	Activating Global DSM.	4.14
	The START Command.	4.14
	Automatic Activation of Global DSM.	4.15
	Deactivating Global DSM.	4.16
	PARASORT.	4.16
	Activating BetterGener.	4.17
	Installing BetterGener with Function BSSIA11.	4.17
	Installing BetterGener with Function BSSIB11.	4.18
	Activating SS11MSG.	4.20
	Installing Release 1.1 Over SyncSort MVS.	4.21
	SMFDSECT Location.	4.21
	Returning SYNCLIST Output and the Installation Tape.	4.21
	Maintenance.	4.22
Chapter 5.	Default Options.	5.1
	Important SyncSort Options.	5.2
	SyncSort Options in SYNCMAC.	5.3
	Setting Tape Sort Defaults.	5.50
	The SYNCLIST Report Program.	5.56
	PARM Values for the Systems Programmer.	5.56
	Changing Default Option Settings.	5.59
Chapter 6.	The BetterGener Utility Program.	6.1
	Eligibility for SyncSort Processing.	6.2
	Installing and Activating BetterGener.	6.2
	Using \$ORTPARM with BetterGener.	6.3
	BetterGener Error Messages.	6.3
Chapter 7.	PARASORT.	7.1
	Creating PARASORT Esoteric Unit Names.	7.1
Chapter 8.	Debugging.	8.1
	Automatic Sort Retry Facility.	8.1
	SyncSort Termination Message.	8.1

Dumps	8.2
The DEBUG Parm Facility	8.2
Error Handling in Global DSM	8.3
Abends in Global DSM	8.3
Contacting Product Services	8.3
Appendix A. Messages	A.1
Global DSM Message Format	A.1
Global DSM Messages	A.2
License Key Messages	A.4
Appendix B. The SyncSort Installation Tape	B.1
Appendix C. Using a Data Set for SyncSort License Keys	C.1
Creating a Key Data Set	C.1
Receiving an E-mailed File	C.3
Updating the Key Data Set	C.4
Index	I.1

SyncSort for z/OS Release 1.1 - Summary of Changes

SyncSort for z/OS is a new product for the IBM z/OS operating system and its underlying 64-bit z/Architecture. SyncSort for z/OS is the successor to SyncSort MVS in the same way that z/OS extends the capabilities of MVS.

Release 1.1 of SyncSort for z/OS was preceded by release 1.0. This Summary of Changes identifies changes for both releases as follows:

- Text without a change bar in the left margin applies to SyncSort for z/OS release 1.0 and identifies differences from SyncSort MVS 3.7.
- Text with a solid bar (|) in the left margin applies to SyncSort for z/OS 1.1 only and identifies differences from release 1.0.
- Text with a dashed bar (|) in the left margin applies to changes introduced in the 1.1C/TPF2 level of release 1.1.

Thus, if you are moving from SyncSort MVS 3.7 to SyncSort for z/OS 1.1, you should read the entire Summary of Changes. If you are moving from SyncSort for z/OS 1.0 to release 1.1, you need only read the text indicated by change bars. If you are moving from the TPF0 or TPF1 level to the TPF2 level of release 1.1, you need only read the text indicated by the dashed change bars.

Note that change bars are used in the Summary of Changes only.

Default Options

- DYNALOC

The default setting for the third parameter of the DYNALOC option has been changed from OFF to ON. ON tells SyncSort to dynamically allocate SORTWK data sets when appropriate. Note that if the defaults for the other DYNALOC parameters are not changed, this new setting can result in the allocation of more disk space for MAXSORTs. For a MAXSORT, SyncSort will dynamically allocate n SORTWK data sets, where n is the number specified in the second DYNALOC parameter, subject to the limits of the MAXWKSP option. If you do not want additional space allocated for a MAXSORT application with JCL-allocated SORTWK files, you can change the fourth DYNALOC parameter to OFF.

The maximum number of SORTWKs, specified on the MAXNUM subparameter, has been increased from 100 to 255.

- KEYDSN

The KEYDSN option has been added. The KEYDSN option informs SyncSort that license keys will come from a separate data set instead of from the KEY parameter.

- KEYMSG

The KEYMSG option has been added. The KEYMSG option controls the destination of key-related warning messages.

- KEYWARN

The KEYWARN option has been added. The KEYWARN option sets the number of days before contract expiration that the WER901I expiration warning message begins to appear.

- NULLOFL

The NULLOFL option has been added. The NULLOFL option specifies the action to be taken when any non-SORTOUT UTFIL data set contains no data records.

- NULLOUT

The NULLOUT option has been added. The NULLOUT option specifies the action to be taken when SORTOUT contains no data records.

- SOPAD

The SOPAD option has been added. The SOPAD option specifies the action to be taken if the LRECL defined in the JCL for a non-UTFIL SORTOUT is larger than the SORTIN/SORTINnn LRECL or the internally processed record length when the SORTIN/SORTINnn LRECL is modified by features.

- SOPADGN

The SOPADGN option has been added. The SOPADGN option specifies the action to be taken in BetterGener applications if the LRECL defined in the JCL for SYSUT2 is larger than the SYSUT1 LRECL or the internally processed record length when the SYSUT1 LRECL is modified by features.

- SOTRN

The SOTRN option has been added. The SOTRN option specifies the action to be taken if the LRECL defined in the JCL for a non-UTFIL SORTOUT is smaller than the SORTIN/SORTINnn LRECL or the internally processed record length when the SORTIN/SORTINnn LRECL is modified by features.

- SOTRNGN

The SOTRNGN option has been added. The SOTRNGN option specifies the action to be taken in BetterGener applications if the LRECL defined in the JCL for SYSUT2 is smaller than the SYSUT1 LRECL or the internally processed record length when the SYSUT1 LRECL is modified by features.

- SUMOVFL

The SUMOVFL option has been added. The SUMOVFL option specifies the action to be taken if a summary field overflows or underflows during SUM processing.

- VLTEST

The OFF4 subparameter has been added to the VLTEST option. OFF4 specifies the action to be taken if an illogical record segment is found.

Authorizing Usage of SyncSort for z/OS

SyncSort for z/OS may only be used by currently licensed customers or on an authorized trial basis. In order to insure licensed usage of the SyncSort product, a valid license key for each authorized machine must be specified. The key can be specified in the KEY option of SYNCMAC or in a separate data set whose name is specified in the KEYDSN option of SYNCMAC.

Using a separate data set eliminates the need to assemble and link modules each time a key is added or replaced and avoids SMP/E involvement. This option is recommended for sites that have many license keys and experience frequent key changes.

Using the data set also allows you to suppress key-related warning messages easily.

Performance Improvements

SyncSort for z/OS exploits the advanced facilities of the zSeries architecture to achieve significant performance improvements:

- Algorithmic improvements have been made to exploit the z/Architecture enhancements of 64-bit central storage support and the elimination of expanded storage. These improvements apply to applications that would have formerly used hiperspace to exploit available expanded storage in a 31-bit OS/390 architecture. A new SyncSort technique, called ZSPACE, allows the native use of the central storage resources without incurring the CPU overhead associated with hiperspace simulation in a 64-bit z/Architecture environment. The technique reduces CPU time and elapsed time. The informational message WER418I indicates if ZSPACE is being used.

- Parallel access volume (PAV) technology, such as on IBM 2105 ESS (SHARK) devices and EMC Symmetrix devices, is exploited to reduce elapsed time.
- Dynamic Storage Management enhancements have been made to exploit the availability of expanded central storage resources found on zSeries servers. Optimization algorithms have been modified to employ additional storage resources when available. This change reduces sorting CPU time, EXCPs, and elapsed time for files that are larger than 600 megabytes.

DB2 Query Support

SyncSort can now directly retrieve data from a DB2 database based on a user-provided query. An SQL SELECT statement is used to specify the criteria of the request. The query of the DB2 database replaces SyncSort's SORTIN or E15 processing. The SORT or COPY functions, but not MERGE, can be used with DB2 queries. All SyncSort features performed after E15 processing are available for use with the DB2 query facility.

This feature improves performance over DB2's DSNTIAUL program by allowing DB2 data to be passed directly into a SORT or COPY operation, without the use of setup steps or the need for user-written E15 exits.

SORTWORK

The maximum number of SORTWK data sets, as specified on the DYNALLOC parameter of the SORT control statement or the DYNALLOC PARM option, has been increased from 100 to 255.

Data Utility Features

The SyncSort for z/OS data utility features have been enhanced with the following:

- INCLUDE/OMIT and OUTFIL INCLUDE/OMIT Statements
 - Fields can now be compared to the date of a SyncSort run or the date of the run with an offset. A variety of forms is available to represent the current date used in the comparison. This allows records to be included or omitted based on whether their dates are equal to, less than, or greater than the run date or the run date with an offset. The forms of the current date constants available for standard comparisons are &DATEX, &DATEX(c), &DATEXP, and Y'DATEX'.
 - Data fields of the formats Y2B, Y2C, Y2D, Y2P, Y2S, and Y2Z can now be compared to a year constant.
 - Data fields that represent the full-date formats Y2T, Y2U, Y2V, Y2W, Y2X, and Y2Y are now supported. Previously these data formats were available for use only with the SORT or MERGE control statement.

- BI (binary) fields can now be compared to positive decimal numbers.
- SORT/MERGE Control Statements
 - The maximum length of an AC or AQ control field has been increased to 4091 bytes (2043 for variable-length records). This is raised from the prior limit of 256 bytes.
- INREC, OUTREC, and OUTFIL OUTREC Statements
 - The date and time of the SyncSort run can now be inserted in different forms in records by the parameters &DATE, &DATE_x, &DATE_x(c), &DATE_xP, &DATE=(m₁m₂m₃m₄), &DATENS=(xyz), &TIME, &TIME_x, &TIME_x(c), &TIME_xP, &TIME=(hp), and &TIMENS=(tt).
 - SMF date and time formats can now be converted to standard date and time formats. The SMF formats are DT1, DT2, DT3, TM1, TM2, TM3, and TM4.
 - The case of EBCDIC letters within a field can now be translated from uppercase to lowercase, from lowercase to uppercase, or the characters in a field can be translated according to an alternate collating sequence (ALTSEQ) table in effect. This is accomplished by using the subparameters TRAN=UTOL, TRAN=LTOU, and TRAN=ALTSEQ, respectively.
 - The conversion of the 2-digit year portion of full-date fields (Y2T, Y2U, Y2V, Y2W, Y2X, and Y2Y) to a 4-digit year in printable format is now supported.
 - A new format parameter, f_{y2b}(c), is now supported. Used with p (position) and l (length) in a p,l,f specification, the new format parameter allows conversion of a full-date field to a printable date with or without separator characters.
 - The new edit pattern M26 has been added.
- OUTFIL Control Statement
 - The VLFILL parameter has been added to the OUTFIL control statement. It is used in conjunction with OUTFIL OUTREC or OUTFIL OUTREC CONVERT to specify a fill byte to be used for any missing p,l field bytes. The VLFILL parameter has two functions:
 - It enables a variable-length OUTFIL OUTREC non-CONVERT application to continue processing when there is an input record with missing field bytes in a p,l field specification. If VLFILL has not been specified, the application will terminate with the critical error WER244A.
 - It provides a means to override the default fill byte used in an OUTFIL OUTREC CONVERT application when there are missing bytes in a p,l field specification. By default, spaces will be used for missing field bytes.

- The NULLOFL option has been added to the OUTFIL control statement. The NULLOFL option specifies the action to be taken when any non-SORTOUT OUTFIL data set contains no data records.
- The FTOV parameter has been added to the OUTFIL control statement. The FTOV parameter converts fixed-length input records to variable-length output records.
- The VLTRIM parameter has been added to the OUTFIL control statement. The VLTRIM parameter defines a byte to be deleted from the end of a variable-length record. All prior occurrences of this byte are deleted until a byte that is not equal to the trim byte is found.
- The REMOVECC parameter has been added to the OUTFIL control statement. The REMOVECC parameter generates reports that do not include ANSI carriage control characters that specify printer actions (for example, skipping a line, ejecting a page). REMOVECC omits the carriage control character from all of the report records.
- The &DATENS=(xyz) and &TIMENS=(tt) parameters, which provide additional formats for inserting the date and time of the SyncSort run in headers and trailers, have been added.

PARM Processing

The following are run-time options for sort, merge, or copy applications. The options do not apply to BetterGener.

Null SORTOUT

- The NULLOUT option has been added. The NULLOUT option specifies the action to be taken when SORTOUT contains no data records. The default is set by the NULLOUT installation option (see "Default Options" above).

Record Padding and Truncation Control

- The PAD option has been added. The PAD option specifies the action to be taken if the LRECL defined in the JCL for a non-OUTFIL SORTOUT is larger than the SORTIN/SORTINnn LRECL or the internally processed record length when the SORTIN/SORTINnn LRECL is modified by features. The default is set by the SOPAD installation option (see "Default Options" above).
- The TRUNC option has been added. The TRUNC option specifies the action to be taken if the LRECL defined in the JCL for a non-OUTFIL SORTOUT is smaller than the SORTIN/SORTINnn LRECL or the internally processed record length when the

SORTIN/SORTINnn LRECL is modified by features. The default is set by the SOTRN installation option (see "Default Options" above).

Sum Processing

- The OVFLO option has been added. The OVFLO option specifies the action to be taken if a summary field overflows or underflows during SUM processing. The default is set by the SUMOVFL installation option (see "Default Options" above).

Record Validity Checking

- The OFF4 subparameter has been added to the VLTEST option. OFF4 specifies the action to be taken if an illogical variable-length record segment is found. The default is set by the VLTEST installation option (see "Default Options" above).

Visual SyncSort for z/OS

SyncSort for z/OS incorporates functionality to integrate Visual SyncSort with SyncSort for z/OS mainframe processing. Visual SyncSort is a separately available PC product that is designed to allow programmers and non-programmers alike to easily create and manage SyncSort applications for the mainframe environment. With Visual SyncSort, you can create new sort, merge, and copy applications, or you can import and modify existing ones. Visual SyncSort saves programmer time while taking full advantage of the processing power of SyncSort for z/OS.

Messages

- Message WER219I has been modified to include an SMS return code provided by SMS in the event of DYNALLOC failure.
- Message WER418I has been modified to indicate whether SyncSort has dynamically chosen to use DataSpace, ZSPACE, or hiperspace during the execution of a sort. See the "Performance Improvements" section above for a description of ZSPACE.
- Message WER456I indicates that a file describing your application has been created and written to the VISUALEX DD statement for export to the PC component of Visual SyncSort.
- Message WER457A indicates that the VISUALEX DD statement for export to the PC component of Visual SyncSort is either missing or its data set has been incorrectly defined.
- Message WER458A indicates that the SYSIN data set created by the PC component of Visual SyncSort cannot be processed by SyncSort due to an insufficient level of maintenance on the SyncSort library.

- Message WER459A indicates that only qualified SyncSort applications may be exported to Visual SyncSort for reasons supplied in the message text.
- Messages WER461I, WER462I, and WER462A support the new installation options SOPAD, SOPADGN, SOTRN, SOTRNGN, and SUMOVL and the run-time equivalents PAD, TRUNC, and OVFL.
- Message WER463A indicates that a linear VSAM data set has been specified on input or output. This type of data set is not supported.
- Message WER464I supports the new OFF4 subparameter of VLTEST.
- Message WER467I indicates that a report of the record layout produced by the DB2 query contained in the SORTDBIN data set has been successfully produced.
- Message WER468A indicates that the DB2 query operation failed and the sort or copy application will not execute. The message text indicates the condition that caused the failure or the DB2 query requirement that was violated.
- Message WER903I indicates that no valid license key was found for use on the specified machine.
- Message WER904I indicates that SyncSort has successfully obtained a valid license key from the key data set whose name is specified in the SYNCMAC installation options macro.
- Message WER905A indicates that SyncSort was unable to obtain a valid license key from the existing key data set due to the reason specified in the message.
- Message WER906I indicates that one or more invalid records were found in the license key data set during an update. Only comment statements, key statements, and valid PARMs statements are permitted in the key data set.
- Message WER907I indicates whether SyncSort may issue certain license key warning messages and is intended to alert you that these warning messages may no longer be posted, though the warning period countdowns will continue.

SMFDSECT

New fields have been added to the SyncSort SMF record. This record is described by the DSECT SMFDSECT, which is placed in the SYNC SORT.R11.CONTROL library during product installation. The new fields are:

- SMFCHDB2. Indicates if application input records were created by the DB2 Query feature.

- SMFIPLBK. A 4-byte binary field that contains SORTIN block size. When the SORTIN block size is greater than 32760, the original 2-byte SORTIN Block Size field SMFIPBLK will contain a value of X'FFFF'.
- SMFOPLBK. A 4-byte binary field that contains SORTOUT block size. When the SORTOUT block size is greater than 32760, the original 2-byte SORTOUT block size field SMFOPBLK will contain a value of X'FFFF'.
- SMFOPSYS. Contains the product name of the operating system on which SyncSort for z/OS is executing.
- SMFOPVER. Contains the product version of the operating system on which SyncSort for z/OS is executing.
- SMFOPREL. Contains the product release of the operating system on which SyncSort for z/OS is executing.
- SMFOPLEV. Contains the product mod level of the operating system on which SyncSort for z/OS is executing.

Chapter 1. Introduction

The *SyncSort for z/OS Installation Guide* explains the procedures for installing and maintaining SyncSort for z/OS.

Interactive Installation

SyncSort for z/OS can be installed and maintained through an interactive process available to z/OS users of TSO/ISPF. Either non-SMP or SMP/E installation and maintenance can be selected.

The interactive process simplifies installation and maintenance by prompting you for the necessary information and by guiding you through each step.

Non-Interactive Installation

SyncSort for z/OS can also be installed and maintained non-interactively with or without the use of IBM's SMP/E program product.

Structure of the Installation Guide

You should refer only to those chapters of the *Installation Guide* which discuss the type of installation and maintenance relevant to your system.

Interactive Installation

If you are installing and maintaining SyncSort for z/OS interactively through TSO/ISPF, read “Chapter 2. SYNCINIT”.

Non-Interactive Installation (non-SMP)

If you are installing and maintaining SyncSort for z/OS non-interactively without the use of SMP, read the following chapters:

- Chapter 3. Installation (Non-SMP)
- Chapter 5. Default Options
- Chapter 6. The BetterGener Utility Program
- Chapter 7. PARASORT

Non-Interactive Installation (SMP/E)

If you are installing and maintaining SyncSort for z/OS non-interactively with the use of SMP/E, read the following chapters:

- Chapter 4. Installation (SMP/E)
- Chapter 5. Default Options
- Chapter 6. The BetterGener Utility Program
- Chapter 7. PARASORT

Problem Solving

If you need help with problems in running SyncSort for z/OS, read “Chapter 8. Debugging”.

Contents of the Installation Tape

The contents of the SyncSort for z/OS installation tape are described in “Appendix B. The SyncSort Installation Tape”.

Installation Considerations

The sections that follow provide an overview of SyncSort facilities you should be aware of before beginning the installation process. The section headings are:

- System Resident or Non-Resident Configuration

- Installing the BetterGener Facility
- PARASORT and Special Esoteric Unit Names
- Dynamic Storage Management (DSM)
- Global DSM Functions

Additional details are provided in Chapters 3 and 4, which describe the non-SMP and SMP installation procedures, respectively.

System Resident or Non-Resident Configuration

Either a system resident or non-resident version of SyncSort can be installed. When the resident version is installed, multiple sorts can share the same SyncSort load modules. Performance can be improved, and overhead can be reduced.

All the resident modules of SyncSort will be installed above the 16-megabyte line. In this way, more space is available in the Link Pack Area for programs which must reside below the 16-megabyte line.

Installing the BetterGener Facility

BetterGener is a high-performance, transparent copy facility, which, in many cases, can replace IEBGENER. When BetterGener is installed, eligible jobs are automatically processed by SyncSort. You do not have to change IEBGENER job streams in order to take advantage of SyncSort's much more efficient copying techniques.

Installing BetterGener will allow your installation to achieve impressive reductions in CPU time and EXCPs.

PARASORT and Special Esoteric Unit Names

PARASORT improves elapsed time performance for sorts whose input is a multi-volume tape data set and/or concatenated tape data sets. Reduced elapsed time can help critical sort applications achieve batch window goals.

The performance improvement from PARASORT is a result of processing the SORTIN input volumes in a parallel fashion. Depending upon the resources provided, elapsed time can be reduced up to 20% for 2-way input and up to 33% for 4-way input.

PARASORT requires additional tape units for the application. You will need from two to eight times the current number of tape units, depending upon resource availability and the degree of improvement desired. PARASORT automatically manages the tape units and

minimizes the use of the tape drive resources by deallocating excess tape drives during initialization and releasing all the extra units at the end of the sort input phase.

The additional tape units are defined to PARASORT on up to four DDs labeled SORTPAR1, SORTPAR2, SORTPAR3, and SORTPAR4. A segment of SORTIN will be read in parallel from each of these DDs. The segmentation of SORTIN is automatic.

Increased parallel input processing (up to four SORTPARn DDs) increases the elapsed time benefit. However, for optimal PARASORT performance, SyncSort must be able to read each SORTPARn input DD simultaneously with no channel contention.

Generally, the normal allocation of tape drives will not ensure sufficient channel path availability, particularly for a 4-way PARASORT (four SORTPARn DD statements). Therefore, you may need to create **special esoteric unit names** for PARASORT. Assigning certain groups of drives to each esoteric unit name used in the PARASORT JCL will in most cases ensure the required channel separation.

Although you may need to create special esoteric unit names, PARASORT itself is included with SyncSort and will be available once SyncSort is installed. If you will be using PARASORT, see the instructions for creating PARASORT esoteric unit names in “Chapter 7. PARASORT”.

Dynamic Storage Management (DSM)

SyncSort's Dynamic Storage Management (DSM) capability is an automatic facility for the dynamic control of memory utilization and SORTWK device selection.

In the case of dynamic control of memory, the DSM facility provides SyncSort with access to both data space and hiperspace and allows SyncSort to dynamically control how much memory (address space, data space, and hiperspace) it uses in all environments. DSM is designed to provide the best performance for each sort while optimizing overall system throughput.

A portion of the address space may be allocated for SyncSort's ZSPACE technique. This technique was created as a replacement for hiperspace. It allows native use of the central storage resources that are available. This technique eliminates the additional overhead produced when hiperspace is simulated by the z/OS operating system in a z/Architecture environment. The ZSPACE technique provides superior CPU performance and reduced system overhead compared to a conventional hiperspace application.

The DSM facility considers the VSCORET option to be an initial recommendation for the amount of address space memory to use. From this starting point, DSM's sophisticated algorithms analyze overall system activity and the particular sort's resource requirements. Using this information, the actual amount of address space and data space to use for the particular sort is determined at run time.

For example, during periods of low activity on the system, the DSM facility makes efficient use of idle memory, providing better performance for the particular sort that's running *and* improving throughput for the entire system. On a busy system (as when an interactive facility is running), SyncSort utilizes less memory so that more remains available for other system tasks.

In the area of dynamic SORTWK device selection, DSM chooses devices from among those designated for its use on the basis of speed and the level of contention for those devices. DSM looks at contention from all applications and from other sorts currently running on the system.

The DSM facility can be operated in one of two ways. The most effective way is through a centralized administrative program that executes in its own address space and communicates with sorts in progress through the z/OS subsystem interface. The term "global DSM" will be used to refer to this preferred way of using DSM throughout this manual.

A second method of using DSM is locally from within a particular execution of the SyncSort product itself. This method will be referred to as "local DSM" and can be used in all operating environments. The resources coordinated by local DSM are a subset of those handled by global DSM, and its decisions about resource usage are based on local rather than global considerations. If global DSM is active, it will supersede the local DSM.

Global DSM Functions

Global DSM has two functions: monitoring and decision-making.

Monitoring

When global DSM is active, it continually monitors the state and performance of SORTWK devices, central and expanded storage, and DASD I/O channel paths. The knowledge it acquires through monitoring is recorded in a special history database, allocated as a single small data set. The information in the database summarizes the patterns and regularities that govern the daily and weekly cycles of system activity. Since monitoring is an ongoing process, the database will always contain an accurate and up-to-date profile of the system.

Decision-Making

Global DSM's second function is to decide how each sort can make the best possible use of available resources. Global DSM's algorithms analyze and coordinate the information in the history database, the current level of resource usage, and the sort's own characteristics. In this way, global DSM selects the best way to balance available system resources and run each sort as efficiently as possible while improving overall system throughput.

Although global DSM considers virtual storage in the sort's address space, data space, hiperspace, and SORTWK, its true focus is on the underlying *real* resources: central stor-

age, expanded storage in non-z/Architecture environments, and available SORTWK disk devices and their associated channel paths.

Local DSM

Local DSM is a variation of SyncSort's DSM facility in which the information used for DSM's decision-making is limited to the information available in the sort's address space. This includes the characteristics of the specific sort and the current availability of system resources, but does not include the system monitoring and history information used by global DSM. The decisions made by local DSM will improve individual sort efficiency and system throughput, but not as effectively as the decisions made by global DSM.

Local DSM is controlled by subparameters of the DSM and DSMWEND options in SYNCMAC. These subparameters are only effective when global DSM is inactive. For more information, see "The DSM and DSMWEND Options" on page 5.8.

Visual SyncSort for z/OS

Visual SyncSort for z/OS is a PC-based product designed to allow programmers and non-programmers alike to easily create and manage SyncSort applications for the z/OS environment. With Visual SyncSort for z/OS, you can create new sort, merge, and copy applications, or you can import and modify existing ones. Visual SyncSort for z/OS saves programmer time while taking full advantage of the processing power of SyncSort for z/OS.

Notational Conventions

The following notational conventions are observed in this manual:

- Braces ({ }) indicate that a choice must be made among the alternatives listed.
- Brackets ([]) indicate an optional item. Two or more items in brackets are mutually exclusive options; only one can be chosen for a particular application.
- Defaults are underlined.
- Upper-case letters and numbers, commas, equal signs, and parentheses must be entered exactly as indicated. Lower-case letters represent variables, which must be replaced by actual values.
- Subscripts show position in a series, and three dots indicate an ellipsis.

For example, a_1, a_2, \dots, a_5 is equivalent to a_1, a_2, a_3, a_4, a_5 and represents five "a" items (variables which will be replaced with actual values).

- Examples, which may be entered exactly as shown, are presented in the Courier typeface as, for instance:

```
ALTSEQ CODE=(F0B7,F1B8,F2B9,F3BA,F4BB,F5BC,F6BD,F7BE,F8BF,F9C0)
```

Related Reading

The following guides supplement the information provided in the *Installation Guide*.

Programmer's Guide

This manual explains how to use SyncSort for z/OS and its control statements.

Reference Guide

This handbook, intended for quick reference, provides the syntax for SyncSort for z/OS control statements and briefly describes each parameter.

Exploiting SyncSort: SortWriter Data Utilities Guide

This two-part user's guide demonstrates how SyncSort's versatile Data Utility features provide an efficient, one-step alternative to writing, testing, and debugging programs. Five comprehensive sample applications illustrate how the control statements work together to produce formatted reports.

Exploiting SyncSort: MAXSORT

This user's guide explains how to use the special MAXSORT feature of SyncSort for z/OS to sort very large amounts of data with only a limited amount of disk space. MAXSORT's unique restart capability is described and sample job control streams and tuning information are included.

Chapter 2. SYNCINIT

SYNCINIT is an interactive installation and maintenance facility. SYNCINIT is designed to allow systems programmers to install and maintain SyncSort interactively with or without SMP/E. SYNCINIT prompts for information and then either builds JCL jobstreams for batch execution in the background or executes IBM utility programs in the foreground.

Note the following requirements and restrictions:

- You cannot maintain SyncSort with SYNCINIT unless you first install SyncSort with SYNCINIT.
- BetterGener can be installed through SYNCINIT, but Tape Sort cannot. If you must use Tape Sort, you must install SyncSort non-interactively.
- If you use SMP/E and you want to install into a pre-existing CSI, you must install SyncSort non-interactively.
- You cannot use SYNCINIT to apply individual APARs supplied to you over the telephone or Internet.

Installing SYNCINIT

SYNCINIT installation consists of three steps: allocating libraries, loading the dialog components, and modifying a CLIST to access SYNCINIT.

STEP 1: Allocate Libraries for SYNCINIT

Use option 3.2 in the ISPF/PDF facility to allocate the five libraries needed to install SYNCINIT. The libraries must be allocated with a RECFM of FB and an LRECL of 80. The following table describes the libraries and gives the number of primary and directory blocks needed for each.

Description	DDNAME	Primary Data Blocks for 6320 BLKSIZE	Directory Blocks
CLIST Library	SYSPROC	60	4
Message Library	ISPMLIB	5	1
Panel Library	ISPPLIB	280	25
Skeleton Library	ISPSLIB	55	3
File Tailoring Library	ISPFIL	50	5

Table 1. SYNCINIT Libraries

Note: Since the SYNCINIT CLISTS are delivered as fixed length records, they may not be compatible with variable length CLIST libraries. If your CLIST libraries are variable length, you should use the non-interactive installation method.

STEP 2: Load Dialog Components

After allocating the libraries required for installing SYNCINIT, submit the JCL shown below to load the dialog components from the SyncSort installation tape to the newly allocated libraries.

Enter the actual names of the newly allocated libraries in the JCL statements labeled LOAD1, LOAD2, LOAD3, LOAD4, and LOAD5.

```
//LOAD      PROC
//          EXEC PGM=IEBCOPY
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=TAPE,VOL=(,RETAIN,SER=SSIZ11),
//          LABEL=(&N,SL),DISP=(SHR,PASS),
//          DSN=SYNCSORT.SYIT.F&N
//SYSUT2   DD   DISP=OLD,DSN=&DSN
//SYSUT3   DD   UNIT=SYSDA,SPACE=(80,(400,1))
//SYSIN    DD   DUMMY
//          PEND
//LOAD1    EXEC LOAD,N=36,DSN='your.CLIST.library'
//LOAD2    EXEC LOAD,N=37,DSN='your.Message.library'
//LOAD3    EXEC LOAD,N=38,DSN='your.Panel.library'
//LOAD4    EXEC LOAD,N=39,DSN='your.Skeleton.library'
//LOAD5    EXEC LOAD,N=40,DSN='your.File.Tailoring.library'
```

STEP 3: Access SYNCINIT Libraries

To access SYNCINIT do the following:

- Review and edit member SE11ALIB in your SYNCINIT CLIST library. You will see complete instructions embedded as comments in the CLIST.
- Place your SYNCINIT CLIST library name in the SYSPROC concatenation of the logon procedure of the person authorized to install and maintain SyncSort.

Acquiring Your License Keys

SyncSort requires a license key for every machine on which it is run. To provide you with a license key, Syncsort Inc. requires a customer profile report form that is produced by the SYNCCPR program. If you do not yet have your license keys, it is a good idea to send the form to Syncsort as quickly as possible. The SyncSort installation process is simplified if you have the keys before you set your default options in SYNCINIT. If you already have your keys, skip this section of the instructions and proceed to “Invoking SYNCINIT” below.

Generating the Customer Profile Report Form

During the download in step 2, a jobstream to create the customer profile report form was placed in member SS11CPR in your ISPFIL library. Review and submit this member. If you will be using SyncSort on multiple machines, submit SS11CPR on each machine.

The customer profile report form contains the serial number from the machine on which SS11CPR was run. Please complete the form with all the information requested. Note that item 5 asks for the machine type and model number. It is important to provide this information if the machine information at the top of the report is incomplete or inaccurate.

After completing a profile form for each applicable machine, please send the forms to Syncsort Inc. The license keys will be generated for you based on the information on the forms.

Invoking SYNCINIT

After dialog components have been loaded into their libraries, you may invoke SYNCINIT from the command line of any ISPF panel, except for the BROWSE/EDIT panels of Version 1 of ISPF/PDF, by entering:

TSO %SYNCINIT

SYNCINIT has a complete tutorial facility that can be accessed by entering T on the initial dialog menu. Additional help is available for each panel when you enter the command HELP or press PF1.

Restoring Delivered Defaults

Any information entered by the installer using SYNCINIT is saved between sessions in the ISPPROF User Profile Library member named SS11PROF. Because only one user is expected to use SYNCINIT, this information is readily available only to that user. If a user with a different USERID wishes to use the dialog, member SS11PROF must be moved to that user's User Profile Library.

If you wish to restore all the delivered default settings after changing them, you can delete or rename member SS11PROF in the ISPPROF User Profile Library.

Chapter 3. Installation (Non-SMP)

This chapter describes how to install SyncSort for z/OS without using IBM's SMP program product. If you are using SMP, see "Chapter 4. Installation (SMP/E)".

Preliminary Tasks

Before beginning to install SyncSort, you must do the following:

1. Prepare for acquiring license keys. SyncSort requires a license key for every machine on which it is run. To provide you with a license key, Syncsort Inc. requires the customer profile report form that is produced by the SYNCCPR program. A jobstream to create this form will be downloaded into your sample JCL library during the first step of the installation process. This jobstream also can be e-mailed to you.

The installation process is simplified if you have the license keys before you set your default options in step 5. If you do not have your license keys already, it is a good idea to send the SYNCCPR output to Syncsort as quickly as possible. If you want to run SYNCCPR before you unload the libraries from the tape, contact Syncsort so that the SYNCCPR jobstream can be e-mailed to you.

The directions for running SYNCCPR are found under "STEP 2: Executing SYNCCPR".

2. Determine the number of the SVC that SyncSort will use.

An SVC allows SyncSort to use its high performance access methods, to optimize the performance of DASD with cache, and to support the key data set facility. SyncSort also uses its SVC to create SMF records.

If you are converting from an earlier release, use a different SVC number for SyncSort than the one you used previously. This will facilitate testing.

See “The SVC Option” on page 5.43 for more information.

3. To use global DSM, ensure there is up to 750K of Extended Common Storage Area (ECSA) available when SyncSort is run. For more information on the DSM facility, see “Dynamic Storage Management (DSM)” on page 1.4.
4. If you have an LE/370 environment, check to see that it is at level 1.4.0 or higher and that you have the LE/370 library named SCEELKED that is required at link-editing time. If so, then you should install the support which gives users the ability to utilize certain SyncSort functions in an LE/370 environment. The functions include:
 - LOCALE processing which allows the user to choose a set of collating rules based on a national language and cultural environment. For more information on LOCALE processing, see “The LOCALE Option” on page 5.17.
 - The processing of E15 and E35 exits written in the C language, COBOL/370, or VS COBOL II. For more information on VS COBOL II exits, see “The COBEXIT Option” on page 5.7.

At execution time, additional subroutines in the LE/370 SCEERUN library must be available to SyncSort.

More information on how to install this support is provided in the JCL that you will unload from the installation tape.

5. If you do not have an LE/370 environment, but intend to write E15 or E35 exits in the C language or VS COBOL II, check to see whether you have the following three IBM libraries:
 - The SEDCBASE library for C.
 - The SIBMBASE library for PL/I.
 - The COB2LIB library for VS COBOL II.

If you have the SEDCBASE and SIBMBASE libraries, then you should install the support for C exits that is appropriate when the LE/370 environment is absent. If you have the COB2LIB, you should also install the support for VS COBOL II exits when an LE/370 environment is absent.

More information on how to install this support is provided in the JCL that you will unload from the installation tape.

6. If you have a DB2 environment, check to see that it is Version 5 or higher and that you have the DB2 library named SDSNLOAD that is required at link editing time. If so,

then you should install the support for specifying DB2 queries to access a DB2 database.

More information on how to install this support is provided in the JCL that you will unload from the installation tape.

7. Decide whether to install Tape Sort.

Tape Sort allows you to use tape SORTWK data sets. Syncsort strongly recommends that you do not install Tape Sort unless you must run jobs using tape SORTWK.

For information on the Tape Sort options, see “Setting Tape Sort Defaults” on page 5.50.

8. Decide whether to activate BetterGener.

The BetterGener utility program offers improved performance over IEBGENER in straight copy tasks and is completely transparent to SyncSort for z/OS users.

More information about BetterGener can be found in “Chapter 6. The BetterGener Utility Program”.

9. Decide if esoteric unit names will be used for PARASORT. PARASORT is installed automatically, but to take advantage of it you may need to create special esoteric unit names. Note that creating PARASORT esoteric unit names need not be done as part of the installation procedure. It can be done at any time. For more information on PARASORT and creating special esoteric unit names for it, see “Chapter 7. PARASORT”.

ELPA Storage Requirements

The following chart lists the storage requirements in the Extended Link Pack Area (ELPA) for a resident configuration of SyncSort. In a non-resident configuration, only the SVC will be in the ELPA.

Location	SVC Module	Sorting Modules
Extended Link Pack Area	40K	1287K

Table 2. ELPA Storage Requirements

DASD Space Requirements

The following chart gives the maximum DASD space needed in blocks:

Data Set Description	Block Size	Primary Data Blocks	Directory Blocks
SYNCLINK (Main LINKLIB Modules)	6144	210	9
SYNCLPA (SVC Module)	6144	16	1
SYNCRENT (Reentrant Sorting Modules)	6144	625	16
SYNCTAPE (Tape Sort Modules)	6144	147	29
SYNCAUTH (Global DSM Modules)	6144	50	4
Global DSM HISTORY	468	6664	0
Intermediate MODLIB	6144	650	60
Intermediate CONTROL	3120	175	3
PPOPTION (Program Product Options)	6320	8	1
Sample JCL	6320	60	4
SYNCLLIB (SS11MSG CLISTS)	6320	6	1
SYNCLMLIB (SS11MSG Messages)	6320	2	1
SYNCLPLIB (SS11MSG Panels)	6320	200	20

Table 3. DASD Space Requirements

Non-SMP Installation Procedure

The procedure for installing SyncSort for z/OS without SMP consists of several steps. It includes suggestions for testing SyncSort and references to the BetterGener activation procedure. These references can help you test BetterGener at the same time you are testing SyncSort, if you wish.

STEP 1: Load Sample JCL Library

Use the following JCL to load the sample JCL library from the SyncSort installation tape.

```
//STEP      EXEC  PGM=IEBCOPY
//SYSUT1    DD    DISP=SHR,DSN=SYNCSORT.RL11.JCL,
//           UNIT=TAPE,VOL=SER=SSIZ11,
//           LABEL=(26,SL)
//SYSUT2    DD    DSN=your.prefix.SYNCSORT.R11.JCL,
//           DISP=(,CATLG),UNIT=SYSDA,VOL=SER=,
//           SPACE=(6320,(60,10,5))
//SYSUT3    DD    UNIT=SYSDA,SPACE=(80,(23,1))
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    DUMMY
```

Some changes are required in the SYSUT2 statement to make the JCL conform to your site's standards.

The output data set will contain these members:

SS11ALOC	SS11GTSI	SS11OPCH
SS11COBL	SS11HALO	SS11SEDC
SS11CPR	SS11HINT	SS11SVEC
SS11DB2	SS11HPRC	SS11TSTF
SS11GENA	SS11INST	SS11TSTI
SS11GENF	SS11ISPA	SS11TVEC
SS11GENT	SS11ISPF	SS11TZAP
SS11GTSF	SS11L370	

Figure 1. Members in Sample JCL Library

STEP 2: Executing SYNCCPR

Before continuing with the installation process, it is a good idea to take the time now to provide the necessary information to acquire your license key. Follow the directions below for executing SYNCCPR and completing the report form, and then continue with the installation process.

In order to generate the appropriate license key, Syncsort Inc. requires certain information about the machine on which SyncSort for z/OS will be running. The SYNCCPR program

prints a report form for this information. Fill out the form and send it to Syncsort for your license key.

Review and submit SS11CPR in the sample JCL library to execute SYNCCPR. If you will be using SyncSort on multiple machines, submit SS11CPR on each machine.

Completing the Report Form

The output of the SYNCCPR program is a customer profile report form. It contains the serial number from the machine on which the program was run. Please complete the form with all the information requested. Note that item 5 asks for the machine type and model number. It is important to provide this information if the machine information at the top of the report is incomplete or inaccurate.

After completing a profile form for each applicable machine, please send the forms to Syncsort. Syncsort will generate license keys for you based on the information on the forms.

STEP 3: Allocate Libraries

To allocate the libraries needed to install SyncSort, edit and submit SS11ALOC, a member in the sample JCL library loaded in step 1. Comments in the member will help you add any information that is needed.

If you are converting from an earlier release of SyncSort, you may want to install into test libraries instead of production libraries to facilitate testing.

STEP 4: Prepare PPOPTION Data Set

Copy members SS11SVEC, SS11TVEC, and SS11TZAP from the sample JCL library loaded in step 1 to the PPOPTION data set allocated in step 3. Use your online copy utility (for example, option 3.3 of ISPF/PDF) to do this.

STEP 5: Choose SyncSort Options

Edit the member SS11SVEC in the PPOPTION data set and change the delivered values for SyncSort's default options. You must change the SVC and KEY or KEYDSN options. Refer to "Chapter 5. Default Options" for detailed information on all of SyncSort's defaults.

If you have decided to install Tape Sort, edit SS11TVEC and SS11TZAP in the PPOPTION data set and change the delivered default values, if necessary. Refer to "Setting Tape Sort Defaults" on page 5.50 for more information on the Tape Sort options.

STEP 6: Process SS11INST

Review, modify, and submit member SS11INST from the sample JCL library.

SS11INST is a JCL jobstream with an inline PROC whose function is to install SyncSort's load modules. Any value that you need to modify is described in the comments within the jobstream.

Once SS11INST is executed, SyncSort's BetterGener module will also be installed in your LINKLIB. If you wish to test BetterGener at the same time you are testing SyncSort, you should review "Activating BetterGener" on page 3.12 and "Chapter 6. The BetterGener Utility Program".

STEP 7: Perform Installation Verification Testing

After SS11INST has completed successfully, review, modify, and submit member SS11TSTI from the sample JCL library to confirm your installation is complete up to this point. If you have chosen the KEYDSN option, create your key data set before running the verification test. For more information, see "Appendix C. Using a Data Set for SyncSort License Keys".

If you are testing BetterGener at the same time as you are testing SyncSort, see "STEP 1: Set Up BetterGener For Testing" on page 3.12.

STEP 8: Allocate Global DSM's History Data Set

Review and submit SS11HALO in the sample JCL library to allocate the history data set that will be used by global DSM.

If SyncSort is running on more than one z/OS system image or logical partition (LPAR), you must allocate a history data set for each. SS11HALO contains descriptive comments, and it may be run as many times as necessary to allocate multiple history data sets.

STEP 9: Initialize GLOBAL DSM's History Data Set

There are two ways to initialize the global DSM (GDSM) history data set, depending upon whether you are a first time user of GDSM or are converting from an earlier release of SyncSort in which you were also using GDSM.

- If you have been using GDSM, then the history data set for a particular system can be initialized by copying its existing history data set to the new file you have allocated in the prior step. You can use the COPY feature of SyncSort to perform this function. This technique for creating the history file will allow the Release 1.1 GDSM to take advantage of the history information developed by the earlier release.
- If you are installing GDSM for the first time, you must initialize the history file you created in the prior step. To perform this initialization you must review and submit SS11HINT on each system where you have allocated a history data set. SS11HINT is a member in the sample JCL library, and it contains descriptive comments.

Note that regardless of which technique you are using, you must either copy or run SS11HINT on each system where you have allocated a history data set. This step is neces-

sary because the history data set needs specific information from the system on which it will be used. If the data set is copied or initialized from a different system, the wrong information will be recorded.

If the system identifier on your system changes, you must reinitialize the history data set on that system by running SS11HINT again. The system identifier is the SID parameter in member SMFPRMxx of SYS1.PARMLIB.

STEP 10: Place PROC into SYS1.PROCLIB

Global DSM cannot be activated until you review and submit SS11HPRC to place the procedure SYNCG110 in SYS1.PROCLIB. SS11HPRC is a member in the sample JCL library, and it contains descriptive comments.

You will activate global DSM as part of the next step.

STEP 11: Perform Production Validation Testing

After the installation verification test (SS11TSTI) has run successfully and global DSM is ready for activation, you should perform a variety of production validation tests. To prepare for this type of testing, you must do the following:

1. If you are using a type 3 SVC number in the range 200-255, then update IEASVCxx in SYS1.PARMLIB with the SVC number being used for SyncSort. If you are using SVC 109 with the alternate routing code, then do not update IEASVCxx.
2. Update LPALSTxx in SYS1.PARMLIB with the SYNCLPA data set name. If you want a system resident configuration of SyncSort, also place the SYNCRENT data set name in LPALSTxx.
3. Update IEAAPFxx in SYS1.PARMLIB with the SYNCAUTH data set name.
4. After updating SYS1.PARMLIB, IPL your system.
5. After the IPL, activate global DSM by issuing the START command for procedure SYNCG110. For complete information, see “Activating Global DSM” on page 3.9.

During production validation testing, you should run the key jobs at your installation in parallel with the current production runs of those jobs, and compare the results. You should also make the new release of SyncSort available to selected users. The jobstreams for both the parallel jobs and the selected user jobs should contain STEPLIBs linking them to your SYNCLINK library. (If you have not chosen a system resident configuration, the jobstreams should also contain a STEPLIB to the SYNCRENT library.)

If you are testing BetterGener at the same time as you are testing SyncSort, see “STEP 2: Test BetterGener” on page 3.12.

STEP 12: Put SyncSort into Production

To put the new release of SyncSort into production, update LNKLS_{Txx} in SYS1.PARMLIB with the SYNCLINK data set name. If you have not chosen a system resident configuration, also place the SYNCRENT data set name in LNKLS_{Txx}.

If you wish, you may use member SS11TSTF in the sample JCL library as a production verification test after you update LNKLS_{Txx}.

Note: It is recommended that you keep the previous production version of SyncSort available for a period of time after the new release is placed in production. The previous release can then be referenced with a STEPLIB, if necessary.

If you were testing BetterGener at the same time as you were testing SyncSort and your tests are complete, you can put BetterGener into production by completing the steps in the procedure for activating BetterGener. Begin with “STEP 3: Provide IEBGENER’s New Name to SYNCGENR” on page 3.13.

If DB2 Query Support was installed, DB2 “BIND ADD” authority is needed to run the first sort on each DB2 database where this support will be used. This will create the DB2 “PLAN” that other jobs will use.

Activating Global DSM

Global Dynamic Storage Management (GDSM) should be installed for optimal sort performance and system throughput. Global DSM consists of a centralized administrative program that executes in its own address space and communicates with sorts in progress through the z/OS subsystem interface.

You installed global DSM in the procedure described in “Non-SMP Installation Procedure” on page 3.4, and the technical details for activating it are given in “The START Command” on page 3.10.

For general information on global DSM, see “Dynamic Storage Management (DSM)” on page 1.4. Information on global DSM messages is located in “Appendix A. Messages” on page A.1, and information about global DSM error handling can be found in “Error Handling in Global DSM” on page 8.3.

If you are at the point of putting Release 1.1 and global DSM (GDSM) into production, and you have been using GDSM with an earlier release, you can recopy the production history file of the earlier release to the history file for Release 1.1. This will provide the most up-to-date profile of your system when you start production with Release 1.1 of GDSM. To copy the data set, follow the guidelines used to create the initial test version, which can be found in “STEP 9: Initialize GLOBAL DSM’s History Data Set” on page 3.7.

The START Command

Global DSM may be activated whenever your system is active, even when SyncSort jobs are executing. However, only SyncSort jobs that begin after global DSM is activated will use it.

To activate global DSM, issue the START command from the console. The START command uses the JCL procedure SYNC110, which was placed in SYS1.PROCLIB in “STEP 10: Place PROC into SYS1.PROCLIB” on page 3.8.

The START command must be issued each time you IPL your system unless you choose automatic activation. See the section “Automatic Activation of Global DSM” on page 3.11.

Note: Since global DSM will be active on your system constantly after it is activated with the START command, you may want to first install global DSM on the same test system where you have installed SyncSort for z/OS, or you may want to restrict the time global DSM is active to the hours when you are testing SyncSort by using the START and STOP commands. However, to receive the full benefit of global DSM once SyncSort is in production, global DSM should be active whenever your system is running.

The format of the START command is:

$\left\{ \begin{array}{l} \text{START} \\ \text{S} \end{array} \right\} \text{SYNC110} \left[\begin{array}{l} \text{,HDSN=name} \\ \text{,LIB=name} \\ \text{,SYSOUT=class} \end{array} \right]$

HDSN=name

The HDSN option allows you to use a history data set that has a name other than SYNC110.HISTORY, which is the delivered default name in SYNC110. This option is useful when you are activating global DSM on several different systems since the data sets on different systems may have different names.

LIB=name

The LIB option allows you to change the default library name that contains the global DSM processing modules. The delivered default name is SYNC110.SYNCAUTH.

SYSOUT=class

The SYSOUT option allows you to change the output class for global DSM's printed output. The delivered default is '*'.

START Command Example

Here is an example of the START command:

```
S SYNCG110,HDSN='ALT.HISTORY.NAME'
```

In this example, global DSM is started (S) and the name of the default history data set (SYNCSORT.R11.HISTORY) in the activating PROC (SYNCG110) is replaced with the name ALT.HISTORY.NAME.

When the initialization is successful, message WER605I is routed to the console.

See the appropriate IBM *System Commands Reference* for your system for the complete syntax of the START command.

Automatic Activation of Global DSM

Global DSM can be activated automatically as part of master scheduler initialization whenever a system is IPLed. To do this, place the START command described above in the COMMNDxx member of SYS1.PARMLIB.

The example below shows how the START command would be specified in COMMNDxx.

```
COM='S SYNCG110'
```

You can add the options described above (HSDN, LIB, and SYSOUT) to the START command in COMMNDxx if you wish.

When the initialization is successful, message WER605I is routed to the console.

Deactivating Global DSM

You may deactivate global DSM by issuing the STOP command from the console, even when a SyncSort job is executing. SyncSort will automatically detect the absence of global DSM and will not attempt to communicate with it. If local DSM is enabled, DSM will continue to function locally.

The format of the STOP command is:

```
{ STOP  
  P } SYNCG110
```

Global DSM will issue the message WER607I to acknowledge that the STOP command has been received and deactivation is in progress. When deactivation is complete, message WER606I is routed to the console.

See the appropriate IBM *System Commands Reference* for your system for the complete syntax of the STOP command.

PARASORT

PARASORT improves elapsed time performance for sorts whose input is a multi-volume tape data set and/or concatenated tape data sets. Reduced elapsed time can help critical sort applications achieve batch window goals.

The performance improvement from PARASORT is a result of processing the SORTIN input volumes in a parallel fashion. Depending upon the resources provided, elapsed time can be reduced up to 20% for 2-way input and up to 33% for 4-way input.

Although you may need to create special esoteric unit names, PARASORT itself is included with SyncSort and will be available once SyncSort is installed. Note that creating PARASORT esoteric unit names need not be done as part of the installation procedure. It can be done at any time.

If you will be using PARASORT, see the instructions for creating PARASORT esoteric unit names in “Chapter 7. PARASORT”.

Activating BetterGener

The BetterGener utility program offers improved performance over IBM’s IEBGENER in straight copy tasks and is completely transparent to SyncSort for z/OS users. More information on BetterGener can be found in “Chapter 6. The BetterGener Utility Program”.

STEP 1: Set Up BetterGener For Testing

After you have completed “STEP 6: Process SS11INST” on page 3.6 in the SyncSort installation procedure, the BetterGener module (SYNCGENR) resides in the SyncSort LINKLIB. You can then set up BetterGener for a test period by applying the zap found in member SS11GENT in the sample JCL library loaded at the beginning of the SyncSort installation procedure. Complete instructions are given in SS11GENT.

After the zap in SS11GENT is applied, you can use member SS11GTSI in the sample JCL library to test that BetterGener is installed correctly up to this point.

STEP 2: Test BetterGener

After you have completed “STEP 11: Perform Production Validation Testing” on page 3.8 in the SyncSort installation procedure, you should run a series of jobs to test BetterGener

before putting it into production. You may use any job that would normally call IEBGENER if you do the following:

- Set up a STEPLIB to access the SYNCLINK library. (If you have not chosen a system resident configuration, the jobstream should also contain a STEPLIB to the SYNCRENT library.)
- Execute PGM=SYNCGENR instead of PGM=IEBGENER.

STEP 3: Provide IEBGENER's New Name to SYNCGENR

After the test period is over, you can begin to put BetterGener into production by deciding on a new name for IBM's IEBGENER module and zapping the name into BetterGener's SYNCGENR module.

To give IEBGENER's new name to SYNCGENR, apply the superzap found in member SS11GENF in the sample JCL library loaded at the beginning of the SyncSort installation procedure. Complete instructions are given in the jobstream.

SS11GENF is set up to use OLDGENER as the new name for IBM's IEBGENER. You may change OLDGENER to any name that cannot already be found through your link list.

STEP 4: Rename IBM's IEBGENER

Rename IBM's IEBGENER to the name you have selected. OLDGENER is the default.

If you are using TSO, issue the following command:

```
RENAME 'SYS1.LINKLIB(IEBGENER)' (OLDGENER)
```

Note: You may receive maintenance from IBM for its IEBGENER after it has been renamed. In this case, change the name OLDGENER (or the name you have chosen) to IEBGENER, apply the maintenance, and then change the name back to OLDGENER (or your choice) again. Alternatively, since you are maintaining IBM's IEBGENER with SMP/E, you can code a USERMOD with a ++RENAME statement.

STEP 5: Assign IEBGENER As Alias

In order that no jobstreams need be changed when BetterGener is in production, SYNCGENR must be given the alias IEBGENER.

Use the member SS11GENA in the sample JCL library to assign IEBGENER as an alias for SYNCGENR. Complete instructions are found within SS11GENA.

SS11GENA uses the linkage editor to assign the alias.

STEP 6: Check Access

After SS11GENA has been run successfully and you have completed “STEP 12: Put SyncSort into Production” on page 3.9 in the SyncSort installation procedure, you should insure that all of the following can be accessed through the link list on your system:

- SYNCGENR with an alias of IEBGENER
- The original IBM IEBGENER through a new name such as OLDGENER
- The SyncSort for z/OS modules

You may use member SS11GTSF in the sample JCL library to confirm that the access to BetterGener is correct.

Activating SS11MSG

SS11MSG is designed to allow anyone who uses SyncSort to have online access to SyncSort’s message texts and explanations. This facility is especially useful when information about a SyncSort message is needed, but the *SyncSort for z/OS Programmer’s Guide* is not available. SS11MSG runs under the ISPF Dialog Management Services. It prompts for a SyncSort message number and then displays the corresponding message text and explanation on the screen.

When job SS11INST in step 6 is executed (see page 3.6), the option exists to initiate the SS11MSG installation process by setting the symbolic parameter ISPF=1. This action copies the SS11MSG dialog components from tape to three target libraries named SYNCCLIB (containing TSO CLISTS), SYNCMLIB (containing user error messages), and SYNCPLIB (containing ISPF panels with SyncSort text).

If job SS11INST is mistakenly executed with ISPF=0, which bypasses the steps that copy the SS11MSG components to the target libraries, jobs SS11ISPA and SS11ISPF should be used. SS11ISPA will allocate the three target libraries, and SS11ISPF will copy the components from tape to the target libraries.

STEP 1: Edit CLIST

Review member SS11MSG2 in the SYNCCLIB library. Instructions are embedded in the member directing you to confirm or edit the settings on two variables.

STEP 2: Add CLIST Library to SYSPROC Concatenation

Add the SYNCCLIB library name to the SYSPROC concatenation of the logon procedure for anyone at your site who uses SyncSort. These are normally applications programmers. Alternatively, simply copy the CLISTS out of the library and into another library that is already in the SYSPROC concatenation.

STEP 3: Perform Installation Verification Test

SS11MSG can be tested by issuing the command TSO SS11MSG from the command line of any ISPF panel. When the main SS11MSG screen appears, it will prompt for the 3-digit number and letter of the message you want explained. There are approximately 350 SyncSort messages. Two simple tests should be performed. First, enter 111A and view the complete 111A explanation displayed on a single screen. Then go back to the main screen to do the second test by entering 416B. The 416B display continues through multiple screens. Check that PF8 scrolls forward through the 416B explanation and PF7 scrolls backward.

STEP 4: Put SS11MSG into Production

Put SS11MSG into production by modifying one of the ISPF/PDF menus. It is suggested that you select either an appropriate menu for non-IBM products or the menu from which users select SDSF, since that is where users will see batch jobs with SyncSort output. If you are unsure of how to modify an ISPF/PDF menu, refer to the appropriate IBM manual.

SMFDSECT Location

The DSECT for SyncSort SMF records can be found in the SYNCSORT.R11.CONTROL library. The member name is SMFDSECT.

Returning SYNCLIST Output and the Installation Tape

The SYNCLIST report program produces a formatted listing of the following:

- All generated options from SS11SVEC
- All Special Customer Zaps (SCZs) applied
- The current maintenance (TPF) level
- System environmental information
- A PARASORT configuration report and esoteric unit name table

SYNCLIST is run automatically in step 6 of the SyncSort for z/OS installation procedure, and two copies of the SYNCLIST report are produced for documentation purposes. Send one copy of this report to SyncSort for z/OS Product Services so that the information will be on file if you need to call.

Fax the report to (201) 930-8284, e-mail it to zos_tech@syncsort.com, or mail it to the following address:

Syncsort Incorporated
50 Tice Boulevard
Woodcliff Lake, NJ 07677

Maintenance

There are four types of SyncSort maintenance:

- Applying periodic maintenance TPFs
- Applying zaps
- Copying modules
- Changing default option settings

The first three items in the list above are described in the following three sections. For details on the last item, see “Changing Default Option Settings for Non-SMP Installations” on page 5.59.

Applying Periodic Maintenance

As a licensed SyncSort for z/OS customer, you will receive maintenance tapes periodically from SyncSort for z/OS Product Services. These tapes are called TPF maintenance tapes, and they normally contain preventive maintenance.

Complete information about how to apply the maintenance will be sent along with the TPF tapes.

Applying Zaps

Occasionally, you may need to apply a zap to your copy of SyncSort apart from periodic maintenance. These zaps normally consist of SPZAP control statements that are applied with the SPZAP Service Aid Program.

Copying Modules

If you need to copy any of SyncSort’s modules and you are using an online copy facility, insure that the copy facility that you use does not disassociate modules from their aliases.

Chapter 4. Installation (SMP/E)

This chapter describes how to install SyncSort for z/OS with IBM's SMP/E program product. If you are not using SMP, see "Chapter 3. Installation (Non-SMP)".

The SMP/E installation procedures will vary depending on whether you are installing into new or old libraries:

- The procedures in this chapter are designed for use with new SMP/E libraries. If you are installing into new SMP/E libraries, you can follow the instructions in this chapter exactly.
- If you will be installing into target and distribution zones that contain the SyncSort MVS product, not every direction in this chapter will be applicable to you. For orientation, you should first read the section "Installing Release 1.1 Over SyncSort MVS" on page 4.21. You can then return to this point and resume reading, understanding that you will have to make decisions based on your particular system.

Required Tape Files

For SMP/E installation, 21 files are needed from the SyncSort installation tape:

File Number	Tape Data Set Name
1	SMPMCS
2	SYNSRT.BSSIZ11.F1
3	SYNSRT.BSSIZ11.F2
4	SYNSRT.BSSIB11.F1
5	SYNSRT.BSSIB11.F2
6	SYNSRT.BSSIC11.F1
7	SYNSRT.BSSIC11.F2
8	SYNSRT.BSSID11.F1
9	SYNSRT.BSSID11.F2
10	SYNSRT.BSSIL11.F1
11	SYNSRT.BSSIL11.F2
12	SYNSRT.BSSIM11.F1
13	SYNSRT.BSSIM11.F2
14	SYNSRT.BSSIM11.F3
15	SYNSRT.BSSIP11.F1
16	SYNSRT.BSSIP11.F2
17	SYNSRT.BSSIT11.F1
18	SYNSRT.BSSIT11.F2
19	SYNSRT.BSSIA11.F1
20	SYNSRT.BSSIA11.F2
21	SYNSRT.SAMPLE.JCL

Table 4. Files Needed for SMP/E Installation

Insure that your security system allows you to read the high level qualifiers of these tape data sets.

Preliminary Tasks

Before beginning to install SyncSort, you must do the following:

1. Prepare for acquiring license keys. SyncSort requires a license key for every machine on which it is run. To provide you with a license key, Syncsort Inc. requires the customer profile report form that is produced by the SYNCCPR program. A jobstream

to create this form will be downloaded into your sample JCL library during the first step of the installation process. This jobstream also can be e-mailed to you.

The installation process is simplified if you have the license keys before you set your default options in step 11. If you do not have your license keys already, it is a good idea to send the SYNCCPR output to Syncsort as quickly as possible. If you want to run SYNCCPR before you unload the libraries from the tape, contact us so that we can e-mail the SYNCCPR jobstream to you.

The directions for running SYNCCPR are found under “STEP 2: Executing SYNCCPR”.

2. Determine the number of the SVC that SyncSort will use.

An SVC allows SyncSort to use its high performance access methods, to optimize the performance of DASD with cache, and to support the key data set facility. SyncSort also uses its SVC to create SMF records.

If you are converting from an earlier release, use a different SVC number for SyncSort than the one you used previously. This will facilitate testing.

See “The SVC Option” on page 5.43 for more information.

3. To use global DSM, ensure there is up to 750K of Extended Common Storage Area (ECSA) available when SyncSort is run. For more information on the DSM facility, see “Dynamic Storage Management (DSM)” on page 1.4.
4. If you do not have an LE/370 environment, but intend to write E15 or E35 exits in the C language or VS COBOL II, check to see whether you have SMP/E Release 1.8.1 or higher and the following three IBM libraries:
 - The SEDCBASE library for C
 - The SIBMBASE library for PL/I
 - The COB2LIB library for VS COBOL II

If you have the SEDCBASE and SIBMBASE libraries, install our SMP function named BSSID11 which installs C exit support when the LE/370 environment is absent. If you have the COB2LIB library, install our SMP function named BSSIC11, which installs VS COBOL II exit support when LE/370 environment is absent.

5. If you have an LE/370 environment, check to see that it is at level 1.4.0 or higher and that you have the LE/370 library named SCEELKED required at link-editing time. If you additionally have SMP/E Release 1.8.1 or higher, then install our SMP function named BSSIL11 which gives users the ability to utilize certain SyncSort functions under LE/370. The functions include:

- LOCALE processing which allows the user to choose a set of collating rules based on a national language and cultural environment. For more information on LOCALE processing, see “The LOCALE Option” on page 5.17.
- The processing of E15 and E35 exits written in the C language, COBOL/370, or VS COBOL II. For more information on VS COBOL II exits, see “The COBEXIT Option” on page 5.7.

At execution time, additional subroutines in the LE/370 SCEERUN library must be available to SyncSort.

6. If you have a DB2 environment, check to see that it is Version 5 or higher and that you have the DB2 library named SDSNLOAD that is required at link editing time. If you additionally have SMP/E Release 1.8.1 or higher, then install the SMP function named BSSIP11 that provides support for specifying DB2 queries to access a DB2 database.

More information on how to install this support is provided in the JCL that is unloaded from the installation tape.

7. Decide whether or not to install Tape Sort.

Tape Sort allows you to use tape SORTWK data sets. Syncsort strongly recommends that you do not install Tape Sort unless you must run jobs using tape SORTWK.

For information on the Tape Sort options, see “Setting Tape Sort Defaults” on page 5.50.

8. Decide whether or not to activate BetterGener.

The BetterGener utility program offers improved performance over IEBGENER in straight copy tasks and is completely transparent to SyncSort for z/OS users.

Complete information about BetterGener can be found in “Chapter 6. The BetterGener Utility Program”.

If you wish to use BetterGener, you must install either function BSSIA11 or BSSIB11. BSSIB11 must be installed in the same zones as the main SyncSort functions. BSSIA11 must be installed in the z/OS operating system zones because it links the system module IEBGENER with the BetterGener module SYNCGENR.

9. Decide whether or not you will use DDEFs. The procedure for either case is described in “STEP 6: Create DDDEFs” and “STEP 7: Create SS11SMP in SYS1.PROCLIB” on page 4.10.
10. Decide if esoteric unit names will be used for PARASORT. PARASORT is installed automatically, but to take advantage of it you may need to create special esoteric unit names. Note that creating PARASORT esoteric unit names need not be done as part of the installation procedure. It can be done at any time. For more information on

PARASORT and creating special esoteric unit names for it, see “Chapter 7. PARASORT”.

SYSMODs

The following table lists the names of the SYSMODs used when installing SyncSort along with a short description of the contents of each:

SYSMOD	Description
BSSIZ11	Function containing the SyncSort base code
BSSIC11	Function containing the code for support of VS COBOL II exits without the LE/370 environment
BSSID11	Function containing the code for support of C exits without the LE/370 environment
BSSIL11	Function containing the code for National Language support (LOCALE), VS COBOL II exits, COBOL/370 exits, and C exits in the LE/370 environment
BSSIP11	Function containing the code for specifying DB2 queries to access a DB2 database
BSSIM11	Function containing the code for the SS11MSG ISPF facility that displays SyncSort message explanations
BSSIT11	Function containing the code for Tape Sort
BSSIB11	Function containing the code for BetterGener in SyncSort zones
BSSIA11	Function containing the code for BetterGener in operating system zones

Table 5. SYSMODs Used for Installation

DDnames for Target and Distribution Libraries

The following is a list of the DDnames needed to install SyncSort with SMP/E and the target and distribution libraries that correspond to them:

DDname	Data Set
SYNCLINK	Target for SyncSort and Tape Sort LINKLIB load modules
SYNCLPA	Target for SyncSort SVC load module
SYNCRENT	Target for SyncSort modules that are reentrant but not necessarily system resident
SYNCAUTH	Target for global DSM load modules
SYNCTAPE	Target for Tape Sort subroutine load modules
SYNCTMAC	Target for installation macros
SYNCCLIB	Target for SS11MSG CLISTs
SYNCLMLIB	Target for SS11MSG ISPF Messages
SYNCLPLIB	Target for SS11MSG ISPF Panels
SYNCLIB	Distribution for load module elements
SYNCMAC	Distribution for macro elements
SYNCSRC	Distribution for source elements
SYNCISP	Distribution for CLIST, message, and panel elements

Table 6. DDNames for Target and Distribution Libraries

ELPA Storage Requirements

The following chart lists the storage requirements in the Extended Link Pack Area (ELPA) for a resident configuration of SyncSort. In a non-resident configuration, only the SVC will be in the ELPA.

Location	SVC Module	Sorting Modules
Extended Link Pack Area	40K	1287K

Table 7. ELPA Storage Requirements

DASD Space Requirements

The DASD space requirements for installing SyncSort will vary depending on the functions that you install. The following chart gives the maximum DASD space needed in data blocks.

DDname	Block Size	Primary Data Blocks	Directory Blocks
SYNCLINK	6144	210	9
SYNCLPA	6144	16	1
SYNCRENT	6144	625	16
SYNCAUTH	6144	50	4
SYNCTAPE	6144	147	29
SYNCTMAC	6320	140	1
SYNCCLIB	6320	6	1
SYNCMLIB	6320	2	1
SYNCPLIB	6320	200	20
SYNCLIB	6144	1600	130
SYNCMAC	6320	140	1
SYNCSRC	6320	8	1
SYNCISP	6320	200	20
DSMHIST	468	6664	0
SMPLTS	6144	16	2
SMPMTS	6320	140	1
SMPPTS	6320	100	20
SMPSTS	6320	8	1
SMPSCDS	6320	50	50
SMPLOG	3200	500	N/A

Table 8. DASD Space Requirements

SMPCSI is a VSAM data set that requires approximately 30 tracks. For a complete definition of the SMPCSI cluster, see member SS11ALOE in the sample JCL library that is loaded in “STEP 1: Load Sample JCL Library” on page 4.8.

The total space needed for the SMPTLIB libraries is 15 cylinders. The SMP DSSPACE entry should be (220,20,100).

SMP/E Installation Procedure

The procedure for installing SyncSort for z/OS with SMP/E consists of several steps. It includes suggestions for testing SyncSort and references to the BetterGener activation pro-

cedure. These references can help you test BetterGener at the same time you are testing SyncSort, if you wish.

STEP 1: Load Sample JCL Library

Use the following JCL to load the sample JCL library from the SyncSort installation tape.

```
//          JOB
//STEP      EXEC  PGM=IEBCOPY
//SYSUT1    DD   DISP=SHR,DSN=SYNSRT.SAMPLE.JCL,
//          UNIT=TAPE,VOL=SER=SSIZ11,
//          LABEL=(21,SL)
//SYSUT2    DD   DSN=prefix.SYNCSORT.R11.JCL,
//          DISP=(,CATLG),UNIT=SYSDA,VOL=SER=,
//          SPACE=(6320,(80,10,6))
//SYSUT3    DD   UNIT=SYSDA,SPACE=(80,(30,1))
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   DUMMY
```

Some changes are required in the SYSUT2 statement to make the JCL conform to your site's standards.

The sample JCL library will contain the following members:

SS11ACPT	SS11GAMV	SS11GBUI	SS11RECV
SS11ALOC	SS11GATF	SS11HALO	SS11SVC
SS11ALOE	SS11GATI	SS11HINT	SS11TSTF
SS11APLY	SS11GBAC	SS11HPRC	SS11TSTI
SS11CPR	SS11GBIN	SS11INIT	USERMODA
SS11DDEF	SS11GBTF	SS11OPTS	USERMODB
SS11GACY	SS11GBTI	SS11PRCA	
SS11GAIN	SS11GBUF	SS11PRCB	

Table 9. Members in Sample JCL Library

STEP 2: Executing SYNCCPR

Before continuing with the installation process, it is a good idea to take the time to provide the necessary information to acquire your license key. Follow the directions below for executing SYNCCPR and completing the report form, and then continue with the installation process.

In order to generate the appropriate license key, Syncsort Inc. requires certain information about the machine on which SyncSort for z/OS will be running. The SYNCCPR program prints a report form for this information. You fill out the form and send it to Syncsort for your license key.

Review and submit SS11CPR in the sample JCL library to execute SYNCCPR. If you will be using SyncSort on multiple machines, submit SS11CPR on each machine.

Completing the Report Form

The output of the SYNCCPR program is a customer profile report form. It contains the serial number from the machine on which the program was run. Please complete the form with all the information requested. Note that item 5 asks for the machine type and model number. It is important to provide this information if the machine information at the top of the report is incomplete or inaccurate.

After completing a profile form for each applicable machine, please send the forms to Syncsort. Syncsort will generate the license keys for you based on the information on the forms.

STEP 3: Allocate Target and Distribution Libraries

To allocate the target and distribution libraries needed to install SyncSort, edit and submit SS11ALOC, which was loaded previously. Comments in the member will help you to add any information that is needed.

If you wish to install BetterGener and test it at the same time you are testing SyncSort, you should review “Activating BetterGener” on page 4.17 and “Chapter 6. The BetterGener Utility Program”. You should then choose a BetterGener installation function (BSSIA11 or BSSIB11). If you choose function BSSIA11, then do “STEP 1: Copy IEBGENER to a Test LINKLIB” on page 4.17

STEP 4: Allocate SMP/E Libraries

Modify and submit member SS11ALOE from the sample JCL library. This job allocates the CSI, LOG, LTS, MTS, PTS, STS, and SCDS data sets.

STEP 5: Initialize SMP/E Libraries

Modify and submit member SS11INIT from the sample JCL library to initialize the newly created SMPCSI data set.

STEP 6: Create DDDEFs

If you want to have SMP/E dynamically allocate any data sets needed during command processing (DDDEFs), modify and submit member SS11DDEF from the sample JCL library. If you do not intend to use DDDEFs, review the note below, skip this step, and be sure to submit SS11PRCB in step 7.

Note: You must use DDDEFs if you intend to install any of the optional SMP SYSMODs named BSSIC11, BSSID11, BSSIL11, or BSSIP11. (See Table 5 on page 4.5.) These

SYSMODs make use of the SMP/E CALLLIBS facility introduced in SMP/E Release 1.8.1. These four SYSMODS cannot be installed with SMP/E Release 7 or earlier.

STEP 7: Create SS11SMP in SYS1.PROCLIB

There are two versions of the JCL-cataloged procedure named SS11SMP. Each version can be installed in SYS1.PROCLIB by submitting either member SS11PRCA or SS11PRCB, which are members in the sample JCL library.

- If you intend to use DDDEFs, then edit and submit member SS11PRCA. This job will create a PROC that does **not** have the DD names defined as JCL statements.
- If you do **not** intend to use DDDEFs, edit and submit SS11PRCB. This job will create a PROC with all the DDNAMES defined as JCL statements.

Note: You must use DDDEFs if you intend to install any of the optional SMP SYSMODs named BSSIC11, BSSID11, BSSIL11, or BSSIP11.

STEP 8: RECEIVE the SyncSort Functions

Modify and submit member SS11RECV from the sample JCL library to RECEIVE the SyncSort functions.

As noted in step 6, there are four optional SYSMODs (BSSIC11, BSSID11, BSSIL11, and BSSIP11) that can only be received with SMP/E Release 1.8.1 or higher.

STEP 9: Supply a Module Name for the SyncSort SVC

Modify and submit SS11SVC from the sample JCL library to supply the SyncSort module name for the SVC. You must have the SVC number for SyncSort to complete this step.

STEP 10: APPLY the SyncSort Functions

Modify and submit member SS11APLY from the sample JCL library to APPLY the SyncSort functions.

If you are installing BetterGener at the same time as SyncSort, note the following: If you are installing function BSSIA11, then do “STEP 2: RECEIVE and APPLY Function BSSIA11” on page 4.17. If you are installing function BSSIB11, then do “STEP 1: RECEIVE and APPLY Function BSSIB11” on page 4.18.

STEP 11: Set and Install the SyncSort Options

The SyncSort options are the values that SyncSort uses as defaults when it executes. To set these options, do the following:

1. To change the delivered default values for any of the SyncSort options, change these values in member USERMODA in the sample JCL library. You must change the SVC and KEY or KEYDSN options. A complete explanation of each option is given in “Chapter 5. Default Options”.
2. If you are installing Tape Sort and you need to change the delivered default values for any of the Tape Sort options, modify member USERMODB in the sample JCL library to reflect these changes. More information on these options can be found in “Setting Tape Sort Defaults” on page 5.50.
3. Modify and submit member SS11OPTS from the sample JCL library to RECEIVE and APPLY USERMODA and USERMODB.

STEP 12: Perform Installation Verification Testing

After SS11OPTS has completed successfully, you should review and submit member SS11TSTI in the sample JCL library to confirm that your installation is complete up to this point. If you have chosen the KEYDSN option, you should create your key data set before running the verification test. For more information, see “Appendix C. Using a Data Set for SyncSort License Keys” on page C.1.

If you are testing BetterGener at the same time you are testing SyncSort, note the following: If you are using function BSSIA11, see the information about SS11GATI in “STEP 3: Test BetterGener” on page 4.17. If you are using function BSSIB11, see the information about SS11GBTI in “STEP 2: Set Up BetterGener For a Test Period” on page 4.18.

STEP 13: Allocate Global DSM's History Data Set

Review and submit SS11HALO in the sample JCL library to allocate the history data set that will be used by global DSM.

If SyncSort is running on more than one z/OS system image or logical partition (LPAR), you must allocate a history data set for each. SS11HALO contains descriptive comments, and it may be run as many times as necessary to allocate multiple history data sets.

STEP 14: Initialize Global DSM's History Data Set

There are two ways to initialize the global DSM (GDSM) history data set, depending upon whether you are a first time user of GDSM or are converting from an earlier release of SyncSort in which you were also using GDSM.

- If you have been using GDSM, then the history data set for a particular system can be initialized by copying its existing history data set to the new file you have allocated in the prior step. You can use the COPY feature of SyncSort to perform this function. This technique for creating the history file will allow the Release 1.1 GDSM to take advantage of the history information developed by the earlier release.

- If you are installing GDSM for the first time, you must initialize the history file you created in the prior step. To perform this initialization you must review and submit SS11HINT on each system where you have allocated a history data set. SS11HINT is a member in the sample JCL library, and it contains descriptive comments.

Note that regardless of which technique you are using, you must either copy or run SS11HINT on each system where you have allocated a history data set. This step is necessary because the history data set needs specific information from the system on which it will be used. If the data set is copied or initialized from a different system, the wrong information will be recorded.

If the system identifier on your system changes, you must reinitialize the history data set on that system by running SS11HINT again. The system identifier is the SID parameter in member SMFPRMxx of SYS1.PARMLIB.

STEP 15: Place PROC into SYS1.PROCLIB

Global DSM cannot be activated until you review and submit SS11HPRC to place the procedure SYNCG110 in SYS1.PROCLIB. SS11HPRC is a member in the sample JCL library, and it contains descriptive comments.

You will activate global DSM as part of the next step.

STEP 16: Perform Production Validation Testing

After installation verification testing is complete on all systems and global DSM is ready for activation, you should perform a variety of production validation tests. To prepare for this type of testing, you must do the following:

1. If you are using a type 3 SVC number in the range 200-255, then update IEASVCxx in SYS1.PARMLIB with the SVC number being used for SyncSort. If you are using SVC 109 with the alternate routing code then do not update IEASVCxx.
2. Update LPALSTxx in SYS1.PARMLIB with the SYNCLPA data set name. If you want a system resident configuration of SyncSort, also place the SYNCRENT data set name in LPALSTxx.
3. Update IEAAPFxx in SYS1.PARMLIB with the SYNCAUTH data set name.
4. After updating SYS1.PARMLIB, IPL your system.
5. After the IPL, activate global DSM by issuing the START command for procedure SYNCG110. For complete information on activating global DSM, see “Activating Global DSM” on page 4.14.

During production validation testing, you should run the key jobs at your installation in parallel with the current production runs of those jobs, and compare the results. You should

also make the new release of SyncSort available to selected users. The jobs should contain STEPLIBs linking them to your SYNCLINK library. (If you have not chosen a system resident configuration, the jobstreams should also contain a STEPLIB to the SYNCRENT library.)

If you are testing BetterGener at the same time you are testing SyncSort, see the following information about setting up a STEPLIB to run a series of BetterGener tests: if you are using function BSSIA11, see “STEP 3: Test BetterGener” on page 4.17; if you are using function BSSIB11, see “STEP 3: Test BetterGener” on page 4.18.

STEP 17: Put SyncSort into Production

To put the new release of SyncSort into production, update LNKSTxx in SYS1.PARMLIB with the SYNCLINK data set name. If you have not chosen a system resident configuration, also place the SYNCRENT data set name in LNKSTxx.

If you wish, you may use member SS11TSTF in the sample JCL library as a production verification program after you update LNKSTxx.

Note: Syncsort recommends that you keep the previous production version of SyncSort available for a period of time after the new release is placed in production. The previous release can then be referenced with a STEPLIB, if necessary.

To put BetterGener into production at the same time as SyncSort, note the following: If you are putting function BSSIA11 into production then do “STEP 4: Put BetterGener into Production” on page 4.18. If you are installing function BSSIB11, then do “STEP 4: Rename IBM’s IEBGENER” on page 4.19.

If SYSMOD BSSIP11 (DB2 Query Support) was installed, DB2 “BIND ADD” authority is needed to run the first sort on each DB2 database where this support will be used. This will create the DB2 “PLAN” that other jobs will use.

STEP 18: ACCEPT the SyncSort Functions

Modify and submit member SS11ACPT from the sample JCL library to ACCEPT the SyncSort functions.

If you are accepting BetterGener at the same time as SyncSort, note the following: You can accept function BSSIB11 by doing “STEP 7: ACCEPT Function BSSIB11” on page 4.20. You may accept BSSIA11 into your z/OS operating system distribution zone, but keep in mind after it is accepted you cannot restore it back to IBM’s original IEBGENER. We have not provided a sample JCL member to accept BSSIA11.

Activating Global DSM

Global Dynamic Storage Management (GDSM) should be installed on z/OS systems for optimal sort performance and system throughput. Global DSM consists of a centralized administrative program that executes in its own address space and communicates with sorts in progress through the z/OS subsystem interface.

You installed global DSM in the procedure described in “SMP/E Installation Procedure” on page 4.7, and the technical details for activating it are given in “The START Command” on page 4.14.

For general information on global DSM, see “Dynamic Storage Management (DSM)” on page 1.4. Information on global DSM messages is located in “Appendix A. Messages” on page A.1, and information about global DSM error handling can be found in “Error Handling in Global DSM” on page 8.3.

If you are at the point of putting Release 1.1 and global DSM (GDSM) into production, and you have been using GDSM with an earlier release, you can recopy the production history file of the earlier release to the history file for Release 1.1. This will provide the most up to date profile of your system when you start production with Release 1.1 of GDSM. To copy the data set, follow the guidelines used to create the initial test version, which can be found in “STEP 14: Initialize Global DSM’s History Data Set” on page 4.11.

The START Command

Global DSM may be activated whenever your system is active, even when SyncSort jobs are executing. However, only SyncSort jobs that begin after global DSM is activated will use it.

To activate global DSM, issue the START command from the console. The START command uses the JCL procedure SYNCG110, which was placed in SYS1.PROCLIB in “STEP 15: Place PROC into SYS1.PROCLIB” on page 4.12.

The START command must be issued each time you IPL your system unless you choose automatic activation. See the section “Automatic Activation of Global DSM” on page 4.15.

Note: Since global DSM will be active on your system constantly after it is activated with the START command, you may want to first install global DSM on the same test system where you have installed SyncSort for z/OS, or you may want to restrict the time global DSM is active to the hours when you are testing SyncSort by using the START and STOP commands. However, to receive the full benefit of global DSM once SyncSort is in production, global DSM should be active whenever your system is running.

The format of the START command is:

$\left\{ \begin{array}{l} \text{START} \\ \text{S} \end{array} \right\} \text{SYNCG110} \left[\begin{array}{l} \text{,HDSN=name} \\ \text{,LIB=name} \\ \text{,SYSOUT=class} \end{array} \right]$
--

HDSN=name

The HDSN option allows you to use a history data set that has a name other than SYNCSORT.HISTORY, which is the delivered default name in SYNCG110. This option is useful when you are activating global DSM on several different systems since the data sets on different systems may have different names.

LIB=name

The LIB option allows you to change the default library name that contains the global DSM processing modules. The delivered default name is SYNCSORT.SYNCAUTH.

SYSOUT=class

The SYSOUT option allows you to change the output class for global DSM's printed output. The delivered default is '*'.

START Command Example

Here is an example of the START command:

<pre>S SYNCG110,HDSN='ALT.HISTORY.NAME'</pre>

In this example, global DSM is started (S) and the name of the default history data set (SYNCSORT.R11.HISTORY) in the activating PROC (SYNCG110) is replaced with the name ALT.HISTORY.NAME.

When the initialization is successful, message WER605I is routed to the console.

See the appropriate IBM *System Commands Reference* for your system for the complete syntax of the START command.

Automatic Activation of Global DSM

Global DSM can be activated automatically as part of master scheduler initialization whenever a system is IPLed. To do this, place the START command described above in the COMMNDxx member of SYS1.PARMLIB.

Here is an example of the way in which the START command would be specified in COMMNDxx:

```
COM= 'S  SYNCG110 '
```

You can add the options described above (HSDN, LIB, and SYSOUT) to the START command in COMMNDxx if you wish.

When the initialization is successful, message WER605I is routed to the console.

Deactivating Global DSM

You may deactivate global DSM by issuing the STOP command from the console, even when a SyncSort job is executing. SyncSort will automatically detect the absence of global DSM and will not attempt to communicate with it. If local DSM is enabled, DSM will continue to function locally.

The format of the STOP command follows.

```
{ STOP }  
{ P    } SYNCG110
```

When the deactivation is complete, message WER606I is routed to the console.

See the appropriate IBM *System Commands Reference* for your system for the complete syntax of the STOP command.

PARASORT

PARASORT improves elapsed time performance for sorts whose input is a multi-volume tape data set and/or concatenated tape data sets. Reduced elapsed time can help critical sort applications achieve batch window goals.

The performance improvement from PARASORT is a result of processing the SORTIN input volumes in a parallel fashion. Depending upon the resources provided, elapsed time can be reduced up to 20% for 2-way input and up to 33% for 4-way input.

Although you may need to create special esoteric unit names, PARASORT itself is included with SyncSort and will be available once SyncSort is installed. Note that creating PARASORT esoteric unit names need not be done as part of the installation procedure. It can be done at any time.

If you will be using PARASORT, see the instructions for creating PARASORT esoteric unit names in “Chapter 7. PARASORT”.

Activating BetterGener

The BetterGener utility program offers improved performance over IBM’s IEBGENER in straight copy tasks and is completely transparent to SyncSort for z/OS users. More information on BetterGener can be found in “Chapter 6. The BetterGener Utility Program”.

You can install BetterGener using function BSSIA11 or function BSSIB11. Using BSSIA11 is the preferred way of installing BetterGener because the installation will be totally within SMP/E. However, BSSIA11 must be installed in the same SMP/E control data sets where the z/OS operating system is installed. If you do not wish to install in these data sets, you must use BSSIB11.

Installing BetterGener with Function BSSIA11

Function BSSIA11 links SyncSort’s BetterGener module (SYNCGENR) with IBM’s IEBGENER module. It also sets up GENERATE as an alias for the original entry point to IBM’s IEBGENER.

STEP 1: Copy IEBGENER to a Test LINKLIB

Modify and submit member SS11GACY from the sample JCL library to copy IBM’s IEBGENER to a test LINKLIB.

STEP 2: RECEIVE and APPLY Function BSSIA11

Locate the cataloged procedure and batch JCL that are used to maintain your z/OS operating system with SMP/E. You will need the information in this procedure to RECEIVE and APPLY function BSSIA11.

After you have located the procedure and JCL, review member SS11GAIN in the sample JCL library. SS11GAIN contains descriptive comments.

STEP 3: Test BetterGener

When you have successfully completed the previous steps in this procedure, you may use member SS11GATI in the sample JCL library as an initial test job to verify that BetterGener is correctly installed up to this point.

After SS11GATI is run successfully and you have completed “STEP 16: Perform Production Validation Testing” on page 4.12, you can run a series of BetterGener tests. You can use any job that would normally call IEBGENER as a test job if you set up a STEPLIB to access the BetterGener test LINKLIB and SyncSort’s LINKLIB.

Set up a STEPLIB to access SyncSort's LINKLIB. (If you have not chosen a system resident configuration, the jobstream should also contain a STEPLIB to the SYNCRENT library.)

STEP 4: Put BetterGener into Production

After your test period is complete, you can put BetterGener into production by modifying and submitting member SS11GAMV in the sample JCL library. This job copies the new IEBGENER module and its alias GENERATE from your test library to SYS1.LINKLIB. The new IEBGENER module contains both SYNCGENR and IBM's IEBGENER module, and this new combined module replaces IBM's original IEBGENER module when it is copied.

You may use member SS11GATF in the sample JCL library to test BetterGener at this point if you have completed "STEP 17: Put SyncSort into Production" on page 4.13.

If you wish to bypass BetterGener processing at any time, execute PGM=GENERATE.

Note: If you receive a PUT tape from IBM with maintenance for its IEBGENER, APPLY the maintenance normally unless the maintenance contains new SMP/E JCLIN. If new SMP/E JCLIN is included, you must re-APPLY function BSSIA11 after APPLYing IBM's maintenance.

Installing BetterGener with Function BSSIB11

Function BSSIA11 is the preferred way to install BetterGener. If you choose not to use function BSSIA11, you may use function BSSIB11. However, installation with BSSIB11 will not be completely within SMP/E.

STEP 1: RECEIVE and APPLY Function BSSIB11

After you have completed "STEP 12: Perform Installation Verification Testing" on page 4.11, modify and submit member SS11GBIN from the sample JCL library to RECEIVE and APPLY function BSSIB11.

STEP 2: Set Up BetterGener For a Test Period

Modify and submit member SS11GBUI from the sample JCL library to RECEIVE and APPLY the initial USERMOD SZBG111, which sets up BetterGener for a test period. You can then use member SS11GBTI in the sample JCL library to verify that BetterGener is installed correctly up to this point and is ready for further testing.

STEP 3: Test BetterGener

After you have completed the previous steps and "STEP 16: Perform Production Validation Testing" on page 4.12, you can test BetterGener before putting it into production. You may use any job that would normally call IEBGENER if you do the following:

- Set up a STEPLIB to access SyncSort's LINKLIB. (If you have not chosen a system resident configuration, the jobstream should also contain a STEPLIB to the SYNCRENT library.
- Execute PGM=SYNCGENR instead of PGM=IEBGENER.

STEP 4: Rename IBM's IEBGENER

After your test period is complete, you can begin to put BetterGener into production by renaming IBM's IEBGENER to a name you have selected. OLDGENER is the default.

If you are using TSO, issue the following command:

```
RENAME 'SYS1.LINKLIB(IEBGENER)' (OLDGENER)
```

Note: You may receive maintenance from IBM for its IEBGENER after it has been renamed. In this case, change the name OLDGENER (or the name you have chosen) to IEBGENER, apply the maintenance, and then change the name back to OLDGENER (or your choice) again. Alternatively, since you are maintaining IBM's IEBGENER with SMP/E, you can code a USERMOD with a ++RENAME statement.

STEP 5: Change Access

Modify and submit member SS11GBUF from the sample JCL library to RECEIVE and APPLY the final USERMOD SZBG112, which does the following:

- Creates an alias of IEBGENER for SyncSort's SYNCGENR.
- Defines OLDGENER as the new name for IBM's IEBGENER in the SYNCGENR load module. You may choose a name other than OLDGENER if you wish.

STEP 6: Check Access

After SS11GBUF has been run successfully and you have completed "STEP 17: Put SyncSort into Production" on page 4.13, you should insure that all of the following can be accessed through the link list on your system:

- SYNCGENR with an alias of IEBGENER
- The original IBM IEBGENER through a new name such as OLDGENER
- The SyncSort for z/OS modules

You may use member SS11GBTF in the sample JCL library to confirm that the access to BetterGener is correct.

Note: You may receive maintenance from IBM for its IEBGENER after it has been renamed. In this case, change the name OLDGENER (or the name you have chosen) to IEBGENER, apply the maintenance, and then change the name back to OLDGENER (or your choice) again.

STEP 7: ACCEPT Function BSSIB11

To ACCEPT function BSSIB11, modify and submit member SS11GBAC from the sample JCL library.

Activating SS11MSG

SS11MSG is designed to allow anyone who uses SyncSort to have online access to SyncSort's message texts and explanations. This facility is especially useful when information about a SyncSort message is needed, but the *SyncSort for z/OS Programmer's Guide* is not available. SS11MSG runs under the ISPF Dialog Management Services. It prompts for a SyncSort message number and then displays the corresponding message text and explanation on the screen.

The SMP/E sysmod name for this facility is BSSIM11. The option exists to receive and apply BSSIM11 when the sample jobs SS11RECV and SS11APLY mentioned in steps 8 and 10 (see page 4.10) are submitted. During apply processing, the BSSIM11 components are moved to the three target libraries SYNCCLIB (containing TSO CLISTS), SYNCMLIB (containing user error messages), and SYNCPLIB (containing ISPF Panels with SyncSort text).

If BSSIM11 is mistakenly not received and applied, go back and install this sysmod using sample jobs SS11RECV and SS11APLY before proceeding with the following steps.

STEP 1: Edit CLIST

Review member SS11MSG2 in the SYNCCLIB library. Instructions are embedded in the member directing you to confirm or edit the settings on two variables.

STEP 2: Add CLIST Library to SYSPROC Concatenation

Add the SYNCCLIB library name to the SYSPROC concatenation of the logon procedure for anyone at your site who uses SyncSort. These are normally applications programmers. Alternatively, simply copy the CLISTS out of the library and into another library that is already in the SYSPROC concatenation.

STEP 3: Perform Installation Verification Test

SS11MSG can be tested by issuing the command TSO SS11MSG from the command line of any ISPF panel. When the main SS11MSG screen appears, it will prompt for the 3-digit number and letter of the message you want explained. There are approximately 350 Sync-

Sort messages. Two simple tests should be performed. First, enter 111A and view the complete 111A explanation displayed on a single screen. Then go back to the main screen to do the second test by entering 416B. The 416B display continues through multiple screens. Check that PF8 scrolls forward through the 416B explanation and PF7 scrolls backward.

STEP 4: Put SS11MSG into Production

Put SS11MSG into production by modifying one of the ISPF/PDF menus. It is suggested that you select either an appropriate menu for non-IBM products or the menu from which users select SDSF, since that is where users will see batch jobs with SyncSort output. If you are unsure of how to modify an ISPF/PDF menu, refer to the appropriate IBM manual.

Installing Release 1.1 Over SyncSort MVS

SyncSort for z/OS Release 1.1 can be installed in target and distribution zones that contain SyncSort for z/OS Release 1.0 or any release of SyncSort MVS. Applying SYSMOD BSSIZ11 of SyncSort for z/OS will delete the prior release. Make sure that the target and distribution names listed in “Table 6. DDNames for Target and Distribution Libraries” on page 4.6 are accessible.

SMFDSECT Location

The DSECT for SyncSort SMF records will be found in the target library associated with DDname SYNCTMAC. The member name is SMFDSECT.

Returning SYNCLIST Output and the Installation Tape

The SYNCLIST report program produces a formatted listing of the following:

- All generated options from SYNCMAC
- All Special Customer Zaps (SCZs) applied
- The current maintenance (TPF) level
- System environmental information
- A PARASORT configuration report and esoteric unit name table.

If you are using the members in the sample JCL library that Syncsort provides, SYNCLIST is executed as part of SS11OPTS for SMP/E. However, if you are not using this sample JCL, you should run SYNCLIST for documentation purposes after SyncSort is installed. The following is a sample SYNCLIST execution:

//STEP1	EXEC	PGM=SYNCLIST
//STEPLIB	DD	DSN=SYNCSORT.R11.SYNCLINK, DISP=SHR
//SYSPRINT	DD	SYSOUT=*, COPIES=2
//SYSLIB	DD	DSN=SYNCSORT.R11.SYNCLINK, DISP=SHR
//	DD	DSN=SYNCSORT.R11.SYNCRENT, DISP=SHR

Figure 2. Sample JCL Stream for SYNCLIST

Complete information about SYNCLIST and about this sample execution can be found in “The SYNCLIST Report Program” on page 5.56.

Two copies of the SYNCLIST report will be produced. Send one copy of this report to SyncSort for z/OS Product Services so that the information will be on file if you need to call.

Fax the report to (201) 930-8284, e-mail it to zos_tech@syncsort.com or mail it to the following address:

Syncsort Incorporated
50 Tice Boulevard
Woodcliff Lake, New Jersey 07677

Maintenance

There are four types of SyncSort maintenance:

- Applying periodic maintenance (TPFs)
- Applying APARs
- Copying modules
- Changing default option settings

The first three items in the above list are described in the following three sections. For details on the last item, see “Changing Default Option Settings for SMP Installations” on page 5.60.

Applying Periodic Maintenance

As a licensed SyncSort for z/OS customer, you will receive maintenance tapes periodically from SyncSort for z/OS Product Services. These tapes are called TPF maintenance tapes, and they normally contain preventive maintenance in the form of PTF SYSMODs.

Complete information about how to apply the maintenance will be sent along with the TPF tapes.

Applying Important APARs

Occasionally, you may need to apply an important APAR to your copy of SyncSort apart from periodic maintenance. These APAR SYSMODs normally consist of SMP MCS statements and SPZAP control statements that are applied using SMP/E.

For example, if you received APAR SY00000, you would enter the APAR and embed it in appropriate JCL. If you are using SyncSort's JCL PROC, you might code the following to APPLY the APAR.

```
//STEP      EXEC SS11SMP
//SMPPTFIN DD *
++APAR(SY00000) .
++VER(Z038) FMID(BSSIZ11) .
++ZAP(QKQ0PCAL) .
  NAME QKQ0PCAL
  VER 0000 00000000
  VER 1111 11111111
  REP 0000 22222222
  REP 1111 33333333
/*
//SMPCNTL   DD *
  SET BDY(GLOBAL) .
  RECEIVE SYSMODS SELECT (SY00000) .
  SET BDY(SYNCTGT) .
  APPLY SELECT(SY00000) .
/*
```

Note: Do **not** ACCEPT SyncSort APARs; only SyncSort PTFs and FUNCTIONS should be ACCEPTed.

Copying Modules

If you need to copy any of SyncSort's modules and you are using an online copy facility, insure that the copy facility you use does not disassociate modules from their aliases.

Chapter 5. Default Options

The SYNCMAC macro is used to select permanent SyncSort defaults at installation time. To make changes after installation, use SYNCMAC again to modify individual options. Options for sorting with tape SORTWK data sets (Tape Sort) are generally set with zaps, and those zaps are described later in this chapter.

The format of the SYNCMAC macro is:

<i>Col 1</i>
blank SYNCMAC option,option,...,option
or
label+blank

Figure 3. SYNCMAC Format

<i>Col 16</i>	<i>Col 72</i>
...,option,option,option,option, option,option,option	X

Figure 4. SYNCMAC Continuation Format

Note that column 1 must be blank on the first statement or contain a label. Labels may be up to 8 characters long, and must be followed by at least one blank. To continue a statement, end the statement with the comma after a completed parameter, place any character in column 72, and start the next option in column 16 on the next statement. Be careful not

to omit the delimiting commas separating the SYNCMAC options. An example of an actual SYNCMAC macro follows:

10	16	18		72
SYNMAC NOLISTT , SVC= (109 , 242) , KEYDSN=SYNCSORT . R11 . KEYS ,				X
RSRVTI=60K ,				X
RC16=ABE				

Figure 5. Sample SYNCMAC Macro

For information on how to use the SYNCMAC macro if you are installing without SMP, see “STEP 5: Choose SyncSort Options” on page 3.6. If you are installing with SMP, see “STEP 11: Set and Install the SyncSort Options” on page 4.10.

Important SyncSort Options

Here is a list of options to which you should pay special attention when you are installing SyncSort for z/OS:

- KEY-related parameters: KEY, KEYDSN, KEYMSG, KEYWARN

SyncSort requires a license key for every CPU on which it is run. These options support the installation of the keys, the use of key data sets, and the control of key-related warning messages.

- SVC

An SVC allows SyncSort to use its high performance access methods, to optimize the performance of DASD with cache, and to support the key data set facility. SyncSort also uses its SVC to create SMF records.

If you are converting from an earlier release, use a different SVC number for SyncSort than the one you used previously. This will facilitate testing.

See “The SVC Option” on page 5.43 for more information.

- MINCORE

The MINCORE option specifies the minimum amount of virtual storage that will be available to SyncSort. Make sure that the MINCORE value is high enough to support all the sorts at your installation.

- VSCORE, VSCORET

The VSCORE option specifies the maximum amount of virtual storage available to SyncSort below the 16-megabyte line.

The VSCORET option specifies the total amount of virtual storage available to SyncSort below and above the 16-megabyte line when SyncSort's Dynamic Storage Management (DSM) facility is inactive. When DSM is active, the setting of the VSCORET option is treated as an initial recommendation.

- DSM, DSMWEND

The DSM and DSMWEND options allow you to set the optimization mode for all the sorts run on your system. You can direct SyncSort to provide the best overall balance of CPU time, elapsed time (wall clock time), and I/O activity, or you can direct SyncSort to minimize the use of one of these three. DSM and DSMWEND also allow you to vary the optimization mode during specific time intervals and to control local DSM.

- RSRVJI, RSRVTI

The RSRVJI and RSRVTI options allow you to specify the amount of virtual storage to be reserved below the 16-megabyte line for JCL-initiated and program-invoked sorts. Usually, it is not necessary to reserve any virtual storage for JCL-initiated sorts. You may, however, want to specify the RSRVTI option to reserve virtual storage for invoked sorts.

SyncSort Options in SYNCMAC

All of the SyncSort options are listed below in alphabetical order along with information about coding them as parameters of the SYNCMAC macro.

Defaults are underlined, and you do not need to code any option for which you are using the delivered default setting. However, all options are coded for documentation purposes in the SYNCMAC skeleton supplied with SyncSort.

A number of the SyncSort options in SYNCMAC can be overridden at run time by using their corresponding execution time parameters (PARMs). See the *SyncSort for z/OS Programmer's Guide* as well as the end of this chapter for details.

The ALTMSG Option

$$\text{ALTMSG} = \left\{ \frac{\text{SYSOUT}}{\text{ccccccc}} \right\}$$

The ALTMSG option allows you to change the message data set DDname that will be used for program-initiated applications. The default name is SYSOUT.

To use a different name, substitute it in place of ccccccc. The name may contain up to 8 characters.

If the DDname specified by ALTMSG is not provided in the JCL for a program-initiated application, then SyncSort will use the DDname specified in the MSGDD option, if available.

The ALTPARM Option

$$\text{ALTPARM} = \left\{ \begin{array}{l} \text{\$ORTPARM} \\ \text{ccccccc} \end{array} \right\}$$

The ALTPARM option allows you to change the alternate parameter data set DDname. The default name is \$ORTPARM.

To use a different name, substitute it in place of ccccccc. The name may contain up to 8 characters.

The ALTSEQ Option

$$\text{ALTSEQ} = (\text{ccpp}_1, \dots, \text{ccpp}_{50})$$

The ALTSEQ option modifies the standard EBCDIC collating sequence for control fields labeled with an AQ format code. Note that the installation modifications will be overridden at execution time if an ALTSEQ control statement is supplied.

The coding format of the installation option is similar to that of the control statement. See the *SyncSort for z/OS Programmer's Guide* for additional details.

cc is the hexadecimal representation of the character to be repositioned in the alternate sequence. pp shows the new position in the EBCDIC sequence for the character represented by cc. Up to 50 ccp combinations may be specified.

The default is the standard EBCDIC collating sequence.

The BRKPTIM Option

$$\text{BRKPTIM} = \left\{ \begin{array}{l} \underline{10} \\ \text{nnnn} \end{array} \right\}$$

The BRKPTIM option sets the minimum amount of time that must elapse during a MAXSORT before operator communication at breakpoints after the SORTIME option OC value has expired. The BRKPTIM option is ignored if the SORTIME option is not specified as 'OC' or 'n,OC'.

Specify the number of minutes of execution time (nnnn) that must elapse before the operator is requested to continue or to terminate the sort.

The default is 10 minutes.

The CENTWIN Option

$$\text{CENTWIN} = \left\{ \begin{array}{c} 0 \\ s \\ f \end{array} \right\}$$

The CENTWIN option defines a sliding or fixed 100-year window that determines the century to which 2-digit year data belongs when processed by SORT, MERGE, INREC, OUTREC, or OUTFIL OUTREC control statements.

The 2-digit year data formats (Y2B, Y2C, Y2D, Y2P, Y2S, and Y2Z) plus the full-date formats (Y2T, Y2U, Y2V, Y2W, Y2X, and Y2Y) work with CENTWIN to treat a 2-digit year value as a 4-digit year. The 2-digit and full-date year data formats can be specified on control statements as follows:

- Use SORT/MERGE control statements to correctly collate 2-digit years that span century boundaries.
- Use INCLUDE/OMIT or OUTFIL INCLUDE/OMIT control statements for correct comparisons involving 2-digit year data formats.
- Use INREC/OUTREC or OUTFIL OUTREC control statements to convert 2-digit years to 4-digit printable output.

In addition, two date formats, Y2ID and Y2IP, are provided for year conversion with INREC/OUTREC and OUTFIL OUTREC. These formats work with CENTWIN to expand a 2-digit year in packed decimal format to a 4-digit year while maintaining the packed decimal format in the output field.

For more information on using the date data formats for SORT/MERGE field specifications and INREC or OUTREC processing, see the *SyncSort for z/OS Programmer's Guide*.

CENTWIN ensures that year data spanning centuries will be sequenced correctly. For example, without CENTWIN processing, an ascending sort/merge would sequence the year 01 before the year 98. With CENTWIN processing, the 01 field could be recognized as a twenty-first century date (2001) and would thus be sequenced after 98 (1998) for an ascending sort.

The CENTWIN option generates either a sliding or fixed century window, depending on which form of CENTWIN is used: CENTWIN=s or CENTWIN=f.

- CENTWIN=s specifies a sliding century window, which automatically advances as the current year changes.

The variable s is a number 0 through 100. This value is subtracted from the current year to set a century-window starting point. For example, in 1996 CENTWIN=20 would create the century window 1976 through 2075. Ten years later in 2006, the century starting year would slide to 1986 (2006 minus 20 = 1986) and the century window would be 1986 through 2085.

The CENTWIN delivered default is s=0, which means the current year is the starting year of a century window.

- CENTWIN=f specifies a fixed century window.

The variable f is a 4-digit year (yyyy) between 1000 and 3000. For example, CENTWIN=1976 establishes a fixed starting year 1976 for the century window 1976 through 2075. This window will not change as the current year changes.

The century window defined by CENTWIN controls processing of year-data. If a 2-digit year field (indicated by Y2B, Y2C, Y2D, Y2P, Y2S, Y2Z, Y2ID, Y2IP, Y2T, Y2U, Y2V, Y2W, Y2X, or Y2Y) has a value less than the last two digits of the century window start year, the year field will be treated as a year in the century following the year of the century window. All other 2-digit years will be treated as in the same century as the century window start year.

For example, consider the century window 1950 through 2049. The 2-digit year fields would be processed as follows:

Two-digit Field	Processed as Year
01	2001
49	2049
50	1950
99	1999

An ascending sort of the above sample data would produce output data in the following sequence:

Two-digit Field	Processed as Year
50	1950
99	1999
01	2001
49	2049

The CMP Option

$$\text{CMP} = \left\{ \begin{array}{c} \text{CPD} \\ \text{CLC} \end{array} \right\}$$

The CMP option specifies the kind of compare operation to be used for sort/merge control fields up to 16 bytes long with the format PD or ZD. CPD uses the compare decimal instruction, and CLC uses the compare logical instruction.

PD or ZD fields longer than 16 bytes automatically default to CMP=CLC.

The delivered default, CMP=CPD, provides a more efficient way of comparing records than CMP=CLC. However, if there is invalid data in the zone portion of a ZD field when CMP=CPD is used, the zone fields will be changed as a result of unpacking. If there is invalid data in either a ZD or a PD field when CMP=CPD is used, an OC7abend may occur.

The COBEXIT Option

$$\text{COBEXIT} = \left\{ \begin{array}{c} \text{COB1} \\ \text{COB2} \end{array} \right\}$$

The COBEXIT option indicates which libraries COBOL E15 and E35 exit routines should use when they are executed.

COBEXIT=COB1 specifies that COBOL exits should use either OS/VS COBOL libraries or no libraries at all.

COBEXIT=COB2, the default, specifies that COBOL exits should use VS COBOL II or COBOL/370 libraries.

VS COBOL II or COBOL/370 run-time library modules typically require more virtual storage than OS/VS COBOL library modules, and the amount of additional storage depends on VS COBOL II or COBOL/370 installation options. When VS COBOL II or COBOL/370 run-time library modules are used, it may be necessary to account for this additional storage by adjusting the b value of the Exit-Name parameter on the MODS statement.

If you install SyncSort with COBEXIT=COB1 and later wish to change from OS/VS COBOL to either VS COBOL II or COBOL/370, you must complete a series of steps. The process includes changing the COBEXIT option to COB2 and installing two additional modules in your SyncSort LINKLIB. For the non-SMP method, follow the steps in “Changing Default Option Settings for Non-SMP Installations” on page 5.59. For the SMP/E method, follow the steps in “Changing Default Option Settings for SMP Installations” on page 5.60.

The DSM and DSMWEND Options

$$\left\{ \begin{array}{l} \text{DSM} \\ \text{DSMWEND} \end{array} \right\} = \left\{ \begin{array}{l} \text{interval} \\ (\text{interval}_1, \dots, \text{interval}_5) \\ (\text{OFF } [\text{subp}_4]) \\ (\underline{0000, 2359, 100, \text{BALANCE}}) \end{array} \right\}$$

interval = (subp₁, subp₂ [,subp₃] [,subp₄])

subp₁ = hhmm

subp₂ = hhmm

subp₃ = $\left\{ \begin{array}{l} \text{,pp} \\ \text{,OFF} \end{array} \right\}$

subp₄ = $\left\{ \begin{array}{l} \text{,BALANCE} \\ \text{,CPU} \\ \text{,ELAP} \\ \text{,IO} \end{array} \right\}$

The DSM and DSMWEND options allow you to control local DSM and to set an optimization mode.

SyncSort provides its best performance when global DSM is used instead of local DSM. For this reason, you should use local DSM only when processing requirements are unusual. For more information about local and global DSM, see “Dynamic Storage Management (DSM)” on page 1.4. In addition, you should change the delivered default that controls local DSM only under special circumstances.

DSM and DSMWEND also allow you to set an optimization mode for SyncSort. Normally, you would use the delivered default setting, which provides for a balance among CPU time, elapsed time, and I/O activity. However, in certain circumstances, you may want to direct SyncSort to minimize one of these resources at all times or during specific time intervals. For example, you can direct SyncSort to minimize elapsed time during the hours when your batch window is in effect and balance the use of all resources throughout the rest of the day.

The DSM (valid Monday through Friday) and DSMWEND (valid on weekends) options have four positional subparameters (subp₁, etc.) If you omit any of the optional subparameters, you must remember to include the positional comma.

The first two DSM and DSMWEND subparameters (subp₁ and subp₂) define a time interval. hh represents an hour value from 00 to 23. mm represents a minute value from 00 to 59. If you are setting parameters for local DSM, the DSM facility will be active for the entire minute specified in mm.

The time interval begins at hhmm of subp₁ and ends at hhmm of subp₂. A maximum of five intervals may be specified, and one interval can cross a day boundary (for example, from 8 p.m. to 8 a.m. on weekdays). Time periods cannot overlap.

The third subparameter (subp₃) controls local DSM when global DSM is inactive. Under z/OS or OS/390, you can set a percentage (from 1 to 100) of the maximum amount of expanded storage that SyncSort will use as hiperspace (pp), or you can set local DSM off (OFF). The delivered default is 100. Specify OFF to disable local DSM.

The fourth subparameter (subp₄) describes the optimization mode that SyncSort will use during the time period specified. This optimization mode will be used regardless of whether local or global DSM is enabled.

The values that can be specified in the fourth subparameter are BALANCE, CPU, ELAP, and IO:

BALANCE	Provides the best overall balance of CPU time, sort elapsed time (wall clock time), and I/O activity to SORTIN, SORTOUT, and SORTWORK.
CPU	Minimizes the CPU time of each sort at the expense of sort elapsed time and I/O activity.
ELAP	Minimizes the elapsed time (wall clock time) of each sort at some expense of CPU time.
IO	Minimizes the amount of I/O activity (EXCPs) that a sort will perform.

Here is an example of how you would direct SyncSort to minimize elapsed time during the hours between midnight (0000) and 6 a.m. (0600) and balance the use of all resources throughout the rest of the day (0601 to 2359) on weekdays (Monday to Friday):

```
DSM= ( (0000,0600,,ELAP) , (0601,2359,,BALANCE) )
```

If you wish to disable local DSM at all times, you can specify OFF in place of the first three subparameters. You can also then specify an optimization mode.

Note: If sorts occasionally need to be performed with an optimization mode other than that defined in the DSM and DSMWEND parameters, you can specify a different optimization mode at execution time.

The DSORGDA Option

```
DSORGDA
```

The DSORGDA option permits the use of a random access data set (DSORG=DA) as input to the sort/merge. Note that dummy records will not be recognized as such but will be sorted with the rest of the SORTIN file. The default is to accept only physical sequential (PS) and partitioned organized (PO) data set numbers as input.

The DYNALOC Option

$$\text{DYNALOC} = (\left[\begin{array}{c} \text{ccccccc} \\ \text{SYSDA} \end{array} \right] \left[\begin{array}{c} ,\text{nn} \\ \underline{3} \end{array} \right] \left[\begin{array}{c} ,\text{ON} \\ ,\text{OFF} \end{array} \right] \left[\begin{array}{c} ,\text{ON} \\ ,\text{OFF} \end{array} \right] \left[\begin{array}{c} \underline{6000000} \\ ,\text{nnnnnnnn} \end{array} \right] \\ \left[\begin{array}{c} \underline{3000000} \\ ,\text{nnnnnnnn} \end{array} \right] \left[\begin{array}{c} \underline{3390} \\ ,\text{nnnn} \end{array} \right] \\ \left[,\text{RETRY} = (\left\{ \begin{array}{c} \text{mm} \\ \underline{00} \end{array} \right\} \left\{ \begin{array}{c} ,\text{nn} \\ ,\underline{00} \end{array} \right\}) \right] \\ \left[,\text{SC} = \text{ssssssss} \right] \left[,\text{MAXNUM} = \left\{ \begin{array}{c} \text{nnn} \\ \underline{32} \end{array} \right\} \right])$$

The DYNALOC option is used to dynamically allocate SORTWK data sets. These data sets are deallocated at the conclusion of all program-invoked sorts, and SyncSort may override specified values when it finds that sorting can be completed more efficiently. Some sort applications that use SUM, OUTREC, OUTFIL, or VSAM SORTOUT and do not provide JCL SORTWORKS may automatically have DYNALOC enabled. Therefore, you should ensure that appropriate options are set at installation so that DYNALOC can be used effectively.

If SMS is not installed or active for temporary DASD work data sets, use ccccccc in the first subparameter to give the generic name of the class from which the sort should dynamically allocate the SORTWK data sets. (Note that VIO is not accepted as a unit device.) SYSDA is the default. If SMS is active, refer to the SC subparameter of DYNALOC described below, at the end of this section.

Use nn in the second subparameter to specify the number of SORTWK files to be dynamically allocated for a MAXSORT. The maximum number allowed is 32, and the default is 3. Note that the MAXNUM subparameter of DYNALOC (see definition below) is not applicable to a MAXSORT application.

ON in the third subparameter tells SyncSort to dynamically allocate SORTWK files when appropriate. Dynamic allocation can be provided even if it is not specified as a PARM or on the SORT control statement. ON is the default. OFF tells SyncSort that no dynamic allocation should be done for a sort.

ON in the fourth subparameter tells SyncSort that, although SORTWK files have been allocated in the JCL, additional SORTWK files should be dynamically allocated. ON is the default. OFF in the fourth subparameter tells SyncSort not to dynamically allocate additional SORTWK files if SORTWK files have been allocated in the JCL. Note that this specification will only be used with MAXSORT.

Use the nnnnnnnn in the fifth subparameter to specify the total amount of primary space to be obtained by dynamic allocation of SORTWK files for MAXSORT applications. The default value for primary space is 6 million bytes.

Use the nnnnnnnn in the sixth subparameter to specify the amount of secondary space to be obtained by dynamic allocation of SORTWK files for MAXSORT applications. The default value for secondary space is 3 million bytes.

In the seventh subparameter, specify the DASD with the smallest track capacity that might be encountered at your installation by a dynamically allocated SORTWK data set. The default is 3390. If, during dynamic allocation, SyncSort receives a device with a track capacity smaller than the one specified, it may ignore the specified device type. Below is a table of the devices and their track capacities:

TYPE	TRACK CAPACITY
3380	47968
3390	56664
9345	46456

Table 10. DASD Device Track Capacity

The RETRY, SC, and MAXNUM subparameters must follow all the other positional subparameters in the DYNALOC option.

Specify **RETRY** to enable SyncSort to perform automatic **DYNALLOC** retry. This facility will attempt to avoid a **SORTWK** capacity-exceeded condition when disk space is not immediately available to satisfy a **DYNALLOC** request. SyncSort will automatically wait a prescribed length of time between retries and will retry a request a specified number of times.

Use the **mm** in the first position in the **RETRY** subparameter to specify the number of times SyncSort is to retry a failed **DYNALLOC** request. The minimum value allowed is 0 and the maximum value is 16. 0, the default, tells SyncSort that automatic **DYNALLOC** **RETRY** is not to be done for a sort.

Use the **nn** in the second position in the **RETRY** subparameter to specify the number of minutes to wait between each **DYNALLOC** request. The minimum value allowed is 0 and the maximum value is 15. 0, the default, tells SyncSort that automatic **DYNALLOC** **RETRY** is not to be done for a sort.

If **SMS** is active to manage temporary work data sets, the **SC** subparameter can be specified to instruct SyncSort to **DYNALLOC** **SORTWK**s from storage class (**STORCLAS**) **ssssssss**. If this parameter is not specified in an **SMS** environment, SyncSort will dynamically allocate devices from the generic class **cccccccc** specified in the first subparameter. Note that an installation written automatic class selection (**ACS**) routine can override the **DYNALLOC** parameter you specify.

The **MAXNUM** subparameter is used to set the maximum number of **SORTWK** files to be dynamically allocated in non-**MAXSORT** applications. Specify a value from 32 to 255. SyncSort will automatically determine the appropriate number of **SORTWK** files to use within the cap specified. Note that the specification of values greater than 32 will increase the need for virtual storage below the 16-megabyte line. This storage is required for system control blocks such as **DCBs**. Since most applications can be supported with the minimum of 32, it may be best to specify at run time, via the **DYNALLOC** parameter, a larger value when it is required.

As noted above, if you want to enable **DYNALLOC** with the default settings, specify **DYNALLOC=(,ON)**. This is the minimum specification to ensure dynamic allocation for **SORTWK** data sets.

The DYNATAPE Option

$\left\{ \begin{array}{l} \text{DYNATAPE} \\ \text{NODYNTPE} \end{array} \right\}$
--

The **DYNATAPE** option instructs **MAXSORT** to dynamically allocate any tape drives needed for its merges. **DYNATAPE** may be used instead of (or as a supplement to) **SORTOUxx DD** statements.

NODYNTPE is the default and suppresses dynamic allocation.

The EMSPACE Option

$$\text{EMSPACE} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \text{nK} \\ \text{0} \\ \text{-} \end{array} \right\}$$

The EMSPACE option instructs SyncSort to reserve the specified amount of virtual storage *below* the 16-megabyte line. The EMSPACE specification is always rounded up to the next page size.

The recommended amount of EMSPACE depends on whether or not you have a disk or tape management system, security package, or CA-1 on your system:

- If you are not using a security package or a tape or disk management system, accept the default EMSPACE value of 0.
- If you are using a tape or disk management system, but not CA-1 or a security package, specify EMSPACE=4K.
- If you are using a security package, but not CA-1, specify EMSPACE=12K.
- If you are using CA-1, with or without a security package, specify EMSPACE=20K.

The EQUALS Option

$$\left\{ \begin{array}{l} \text{EQUALS} \\ \underline{\text{NOEQUALS}} \end{array} \right\}$$

The EQUALS option affects the order in which equal-keyed records are processed. Equal-keyed records are records with identical values in all key fields.

If you specify NOEQUALS, the output sequence of the records within a set of equal-keyed records will be unpredictable. NOEQUALS is the default.

If you specify EQUALS, the records within a set of equal-keyed records will be released in the same order, relative to each other, in which they entered the sort or merge.

Because preserving the order of equal-keyed records adds overhead to sort and merge processing, you should only specify EQUALS when necessary.

If a single file is sorted and the EQUALS option has been chosen, the order of the records with equal key fields will be the same in the output file as in the input file.

If EQUALS is chosen for a sort or a merge and there is more than one input file, equal-keyed records from individual input files will appear in contiguous sets in the output file. These sets will be in the order in which their respective input files were defined to the application.

If you are summing records with SyncSort during a particular application and you wish to guarantee that the record preserved is the first record originally released for processing, you can use the EQUALS parameter at execution to override the default set at installation.

The EXITNAME Option

EXITNAME

The EXITNAME option allows you to link-edit an exit routine in SYSIN when the CSECT of the exit routine is not the same as the member name on the MODS control statement.

To allow this type of link editing, specify EXITNAME. The default is not to allow this form of link editing by not specifying EXITNAME.

The IOERR Option

$\left\{ \begin{array}{l} \text{IOERR} = \left\{ \begin{array}{l} \text{ABE} \\ \text{NOABE} \end{array} \right\} \\ \text{NOIOERR} \end{array} \right\}$

The IOERR option controls error handling when there is an I/O error.

If IOERR=ABE is specified and an I/O error occurs, SyncSort will issue user abend 999 instead of a return code 16 or a user abend 16. A WER061A error message will also be routed to the console in addition to the normal destination for messages.

The default is IOERR=NOABE (or NOIOERR). If one of these is specified, SyncSort will terminate with a return code of 16 when I/O errors occur if RC16=NOABE has also been specified. Otherwise, the sort will issue a user abend 16.

The KEY-Related Options (KEY, KEYDSN, KEYMSG, and KEYWARN)

A license key must be provided for each authorized machine on which SyncSort is used.

To obtain a license key, you will need to run the SYNCCPR program. For more information, see “STEP 2: Executing SYNCCPR” on page 3.5 for non-SMP installations or “STEP 2: Executing SYNCCPR” on page 4.8 for SMP installations.

SyncSort license keys can be maintained either in the SYNCMAC installation options macro or in a separate data set. To maintain the keys in SYNCMAC, use the SYNCMAC KEY option. To maintain the keys in a separate data set, specify the name of the data set in the KEYDSN option of SYNCMAC. Using a data set eliminates the need to assemble and link modules each time a key is added or replaced and avoids SMP/E involvement. The data set option is recommended for sites that have many license keys and experience frequent key changes.

The data set can also be used to tell SyncSort to suppress key-related warning messages. As a license nears expiration, or if there has been a change to a machine on which SyncSort is executed, WER901I and WER903I messages may be issued. If you want to suppress these messages, you can do so by using a PARMs statement in the key data set.

The SYNCMAC options KEY, KEYDSN, KEYMSG, and KEYWARN are described below. For information on creating and updating the key data set, see “Appendix C. Using a Data Set for SyncSort License Keys” on page C.1.

The KEY Option

```
KEY = (xxxxxxxxxxxxxxxxx1...,xxxxxxxxxxxxxxxxx14)
KEY2 = (xxxxxxxxxxxxxxxxx15...,xxxxxxxxxxxxxxxxx28)
KEY3 = (xxxxxxxxxxxxxxxxx29...,xxxxxxxxxxxxxxxxx42)
KEY4 = (xxxxxxxxxxxxxxxxx43...,xxxxxxxxxxxxxxxxx56)
...
KEY10 = (xxxxxxxxxxxxxxxxx127...,xxxxxxxxxxxxxxxxx140)
```

The KEY option allows you to specify your license keys in SYNCMAC. Each key is an encrypted sequence of 16 hexadecimal characters.

Multiple keys may be coded to validate one copy of SyncSort for use on up to 140 machines. Because of IBM Assembler language limitations, only 14 keys may be coded in the KEY parameter. The KEY2 through KEY10 parameters should be used in numerical order for additional keys, leaving no unspecified key subparameters before the last one.

Here is an example of the KEY option for a user with three licensed machines:

```
KEY=(123456789ABCDEF0,23456789ABCDEF01,3456789ABCDEF012)
```

The KEY2 parameter should not be used until at least fifteen license keys need to be entered.

The KEYDSN Option

`KEYDSN=data set name`

The KEYDSN option specifies the name of a data set that you will allocate and populate with your license keys.

Note that every sort step needs to have READ access to this data set. WRITE authority should be limited to the systems programming staff that maintains SyncSort. SyncSort dynamically allocates the data set, so there is no need for a DD statement in the sort JCL.

Following is an example of a KEYDSN coding:

`KEYDSN=SYNCSORT.R11.KEYS`

For more information on how to create key data sets, see “Appendix C. Using a Data Set for SyncSort License Keys” on page C.1

Note that since the KEY and KEYDSN parameters are mutually exclusive, you will have to delete the KEY parameter if you currently have it specified among your default options. If you already have SyncSort installed and are changing options at this point, see “Changing Default Option Settings” on page 5.59 for more information.

The KEYMSG Option

$$\text{KEYMSG} = \left(\begin{bmatrix} \text{SYSOUT} \\ \text{NOSYSOUT} \end{bmatrix} \begin{bmatrix} \text{,WTO} \\ \text{,NOWTO} \end{bmatrix} [\text{n1}, \dots, \text{n16}] \right)$$

The KEYMSG option controls the destination of certain key-related warning messages.

If SYSOUT, the default, is specified, the WER901I expiration warning message and the WER903I unlicensed machine warning message are written to the SYSOUT data set. If NOSYSOUT is specified, the writing of these messages to SYSOUT is suppressed, except for the last 7 days of the warning.

If WTO, the default, is specified, these messages are written to the console with a WTO macro several times a day. If NOWTO is specified, the writing of these two messages as WTOs is suppressed, except for the last 7 days of the warning.

The n1 through n16 parameters are optional routing codes to be used when these messages are written to the console with a WTO macro. 16 is the maximum value for each routing code. The default is to issue the WTO without a routing code.

The KEYWARN Option

KEYWARN=60

The KEYWARN option sets the number of days before contract expiration that the WER901I expiration warning message will begin appearing.

The default value is 60. The valid range is from 0 to 180.

The LISTJ Option

$\left\{ \begin{array}{l} \underline{\text{LISTJ}} \\ \text{NOLISTJ} \end{array} \right\}$

The LISTJ option controls whether or not header lines and user control statements are written on the message data set for all JCL-initiated sorts/merges.

Specify LISTJ (the default) if you want these lines and control statements to be written. Specify NOLISTJ if you do not want them to be written.

The LISTT Option

$\left\{ \begin{array}{l} \underline{\text{LISTT}} \\ \text{NOLISTT} \end{array} \right\}$

The LISTT option controls whether or not header lines and user control statements are written on the message data set for all program-invoked sorts and merges.

Specify LISTT (the default) if you want these lines and control statements to be written. Specify NOLISTT if you do not want them to be written.

The LOCALE Option

$\text{LOCALE} = \left\{ \begin{array}{l} \underline{\text{NONE}} \\ \underline{\text{CURRENT}} \\ \text{name} \end{array} \right\}$

The `LOCALE` option controls cultural environment processing, allowing you to choose an alternative set of collating rules based on a specified national language. For `SORT/MERGE` processing, the alternative collating applies to character (CH) fields. For `INCLUDE/OMIT` comparison processing, the alternative collating applies to character fields and hexadecimal constants compared to character fields.

SyncSort employs the callable services of IBM's Language Environment for z/OS to collate data in a way that conforms to the language and conventions of a selected locale. A locale defines single and multi-character collating rules for a cultural environment. Numerous pre-defined locales are available.

`NONE`, the default setting for `LOCALE`, results in normal EBCDIC collating.

`CURRENT` directs SyncSort to use the locale active when SyncSort begins.

name is the name of a supplied or user-defined locale that is to be active during SyncSort processing. A locale name may be up to 32 characters and is not case sensitive. The locale that is active just before SyncSort processing begins will be restored when SyncSort processing completes. The following is a list of locales provided with the IBM National Language Resources Feature of LE/370.

Locale Name	Language	Country
C		
DA_DK	Danish	Denmark
DE_CH	German	Switzerland
DE_DE	German	Germany
EL_GR	Greek	Greece
EN_GB	English	United Kingdom
EN_JP	English	Japan
EN_US	English	United States
ES_ES	Spanish	Spain
FI_FI	Finnish	Finland
FR_BE	French	Belgium
FR_CA	French	Canada
FR_CH	French	Switzerland
FR_FR	French	France
IS_IS	Icelandic	Iceland
IT_IT	Italian	Italy
JA_JP	Japanese	Japan
NL_BE	Dutch	Belgium
NL_NL	Dutch	Netherlands
NO_NO	Norwegian	Norway
PT_PT	Portuguese	Portugal
SV_SE	Swedish	Sweden
TR_TR	Turkish	Turkey

Table 11. Defined Locales

Notes:

1. Make sure the JCL gives SyncSort access to the library that contains the loadable locale routines. For the supplied locales, these are the dynamically loadable routines in

the IBM AD/Cycle LE/370 library. For more information, see the IBM publication *Language Environment for z/OS & VM Installation and Customization Guide, SC26-4817*.

2. If locale processing is used for fields specified in a SORT or MERGE control statement, VLTEST=1 will be forced on in addition to any other VLTEST options in effect. VLTEST=1 will cause SyncSort to terminate if a variable length input record does not contain all SORT/MERGE control fields.
3. Although locale processing can improve performance compared to external collating routines, it should be used only when necessary. Locale processing can significantly degrade SORT/MERGE and INCLUDE/OMIT performance compared to normal collating.
4. An E61 exit cannot be used with locale processing.
5. Locale processing requires additional main storage to support the use of Language Environment facilities. For those jobs that use locale, the below-the-line region size should be increased by 1MB to accommodate the storage needs of the IBM Language Environment modules.
6. A single ampersand character (&) is not accepted by IEV90 and ASMA90 assemblers. If you want an ampersand in a locale name, use two ampersands (&&).

The MAXMERG Option

$$\text{MAXMERG} = \left\{ \begin{array}{l} n \\ 8 \\ (n, \text{OC}) \\ \text{OC} \end{array} \right\}$$

The MAXMERG option limits the number of tape drives that a MAXSORT will dynamically allocate for merging strings of data.

The value specified for n is the maximum number of tape drives that the program will use for merging the intermediate output tapes created on the SORTOUxx tape drives. Any SORTOUxx tape drive allocated in the JCL will be included in n. At least 2 must be specified, and 8 is the default.

Specify 'OC' if you want the operator to choose the maximum number of tape drives at each breakpoint.

Give a value for n *and* specify OC if you want the operator to be able change the value for n at breakpoints.

Note: Enabling operator communications (OC) requires that the operators are familiar with MAXSORT and prepared to respond to MAXSORT messages. See the *SyncSort for z/OS Programmer's Guide* for more information on MAXSORT.

DYNATAPE must be in effect for MAXMERG to take effect.

The MAXWKSP Option

$$\text{MAXWKSP} = \left\{ \begin{array}{l} \text{MAX} \\ \text{nM} \\ \text{n} \end{array} \right\}$$

The MAXWKSP option gives the maximum amount of disk SORTWK space that SyncSort will use to form sorted strings of data during a MAXSORT.

When the default value, MAX, is used, SyncSort will obtain all available secondary allocations for the SORTWK data sets. This will ensure an efficient MAXSORT execution since it will result in the lowest number of individual sorts and merges required to sort the entire input file.

Use nM when you wish to give a specific maximum amount of disk space (n) in megabytes.

Use n when you wish to give a specific maximum amount of disk space in cylinders.

Note: MAXWKSP should be specified as greater than or equal to MINWKSP.

The MINCORE Option

$$\text{MINCORE} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \text{nK} \\ \underline{245760} \end{array} \right\}$$

The MINCORE option specifies the minimum virtual storage request to be honored by SyncSort. Any virtual storage requests less than MINCORE will be raised to this value during SyncSort execution.

Use nnnnnnnn to give a minimum virtual storage value in bytes. Use nK to give a minimum virtual storage value (n) in kilobytes.

The default value is 245,760 bytes (240K). The minimum allowable value is 204,800 bytes (200K).

Storage requirements increase as the quantity and/or complexity of the control statements increase. For example, applications which use SortWriter features or produce multiple output files may require additional storage. Make sure that the MINCORE value is set high enough to support such applications.

The MINMERG Option

$$\text{MINMERG} = \left\{ \begin{array}{c} 2 \\ n \end{array} \right\}$$

The MINMERG option allows you to specify the minimum number of tape drives that a MAXSORT must have for merging intermediate sorted output. The default and minimum permissible is 2.

MAXSORT forces termination when the minimum number of tape drives specified is not available. The MAXSORT may be restarted later when more drives are available.

The MINWKSP Option

$$\text{MINWKSP} = \left\{ \begin{array}{c} n \\ nM \\ 8 \end{array} \right\}$$

The MINWKSP option specifies the minimum amount of SORTWK disk space that must be present for MAXSORT to begin processing. This option helps ensure that each individual sort in a MAXSORT will be large enough so that the input file can be divided into a reasonably small number of segments.

Use *n* to specify the minimum acceptable amount of SORTWK disk space in cylinders.

Use *nM* to specify the minimum acceptable amount of SORTWK disk space (*n*) in megabytes.

The default value is 8 cylinders.

Note: MINWKSP should be specified as less than or equal to MAXWKSP.

The MODSLNK Option

$$\text{MODSLNK} = \left\{ \begin{array}{c} \text{T} \\ \text{N} \end{array} \right\}$$

The MODSLINK option controls exit link editing. When a MODS statement does not specify a link-editing code, the setting of the MODSLNK option determines whether or not the exit should be link edited along with other routines specified for the same sort/merge phase. If you specify MODSLNK=N, SyncSort will not link-edit exits defined on a MODS statement if the link-editing code on the MODS statement is omitted.

The delivered default (MODSLNK=T) instructs SyncSort to dynamically link-edit exits defined on a MODS statement if the link-editing code on the MODS statement is omitted.

Note: The IBM Data Facility Removable Media Manager (RMM) has some applications which use a MODS statement without a link-editing option. In these cases, link editing is not desired.

The MSG Option

$$\text{MSG} = \left\{ \begin{array}{c} \text{AB} \\ \text{AC} \\ \text{AP} \\ \text{CB} \\ \text{CC} \\ \text{CP} \\ \text{NO} \\ \text{PC} \\ \text{SC} \\ \text{SP} \end{array} \right\}$$

The MSG option allows you to indicate where messages are to be routed.

AB causes all messages to be routed both to the message data set and to the console.

AC causes all messages to be routed to the console but none to the message data set.

AP causes all messages to be routed to the message data set but none to the console. This is the default.

CB causes only critical messages to be routed both to the message data set and to the console.

CC causes only critical messages to be routed to the console but none to the message data set.

CP causes only critical messages to be routed to the message data set but none to the console.

NO causes no messages to be routed to either the message data set or to the console.

PC causes all messages to be routed both to the message data set and to the console.

SC causes only critical messages to be routed to the console but all messages to the message data set.

SP causes only critical messages to be routed to the message data set but all messages to the console.

Note that if no messages are to be routed to the message data set (options AC, CC, and NO), the SyncSort header lines and user control statements will also not be written. This condition is handled differently for BetterGener. See “The MSGGENR Option” on page 5.25.

The MSGCLAS Option

MSGCLAS = ([A] [,B] [,C] [,D] [,E] [,F] [,G] [,H] [,I])

The MSGCLAS option defines the types of SyncSort messages to be issued. The message type is determined by the character at the end of the message number.

If this option is not specified, message types A, C, D, and I are the default. Type A messages are critical error messages, and types B, C, D, and I are informational messages. (Note that only types A, B, C, D, and I are meaningful at this time.)

The MSGDD Option

MSGDD = { ccccccc
SYSOUT }

The MSGDD option is used to specify the DDname of the message data set. The default name is SYSOUT, or you can give a specific name (ccccccc).

The MSGGENR Option

$$\text{MSGGENR} = \left\{ \begin{array}{c} \text{AB} \\ \text{AC} \\ \text{AP} \\ \text{CB} \\ \text{CC} \\ \text{CP} \\ \text{NO} \\ \text{PC} \\ \text{SC} \\ \text{SP} \end{array} \right\}$$

The MSGGENR option allows you to indicate where messages are to be routed for BetterGener applications.

See the MSG option for a description of the parameters.

Note that the defaults for MSG and MSGGENR are different. Also, the SyncSort header lines will never be written to the message data set (SYSPRINT) for BetterGener applications.

The MXROUTC Option

$$\text{MXROUTC} = \left\{ \begin{array}{c} (n_1, \dots, n_{16}) \\ \underline{2} \end{array} \right\}$$

The MXROUTC option is used to specify the route code(s) for MAXSORT WTO messages. Acceptable values for n are 1 to 16. The default is route code 2.

The NOSNAP Option

NOSNAP

The NOSNAP option instructs SyncSort not to produce a SNAP dump in the event of an abend during a sort or merge, unless the DEBUG parameter is specified.

The NULLOFL Option

$$\text{NULLOFL} = \left\{ \begin{array}{c} \text{RC0} \\ \text{RC4} \\ \text{RC16} \end{array} \right\}$$

The NULLOFL option specifies the action to be taken when any non-SORTOUT OUTFIL data set contains no data records.

- RC0** The delivered default instructs SyncSort to issue a return code of 0 if not overridden by a higher return code set for another reason.
- RC4** Instructs SyncSort to issue a WER461I warning message and continue processing. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- RC16** Instructs SyncSort to issue a WER461A message and to terminate processing with a return code of 16.

The NULLOUT Option

$$\text{NULLOUT} = \left\{ \begin{array}{c} \text{RC0} \\ \text{RC4} \\ \text{RC16} \end{array} \right\}$$

The NULLOUT option specifies the action to be taken when SORTOUT in a sort, merge, or copy application contains no data records.

- RC0** The delivered default instructs SyncSort to issue a return code of 0 if not overridden by a higher return code set for another reason.
- RC4** Instructs SyncSort to issue a WER461I warning message and continue processing. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- RC16** Instructs SyncSort to issue a WER461A message and to terminate processing with a return code of 16.

The OUTSPCE Option

$$\text{OUTSPCE} = \left\{ \begin{array}{c} \text{nnnn} \\ \underline{16} \\ \text{nnn}\% \\ 0 \end{array} \right\}$$

The OUTSPCE option designates the amount of space for each secondary extent that SyncSort will obtain for the output data set(s) if it has not been specified in the JCL.

You can give a specific amount of space (nnnn) for each secondary extent. nnnn will be in tracks, cylinders, or blocks, depending on the type of space specified on the DD statement(s).

If this option is not specified, the default is 16.

Specify nnn% (a value followed by a percent sign) to give a percentage of primary space to be used for each secondary extent in the output data set(s).

Specify 0 to signal that no automatic secondary space is to be obtained for the output data set(s).

The PARMEXIT Option

PARMEXIT

The PARMEXIT option controls whether or not the PARMEXIT facility can be used. To enable the PARMEXIT facility, specify PARMEXIT. The default is not to specify PARMEXIT, and consequently to disable this facility.

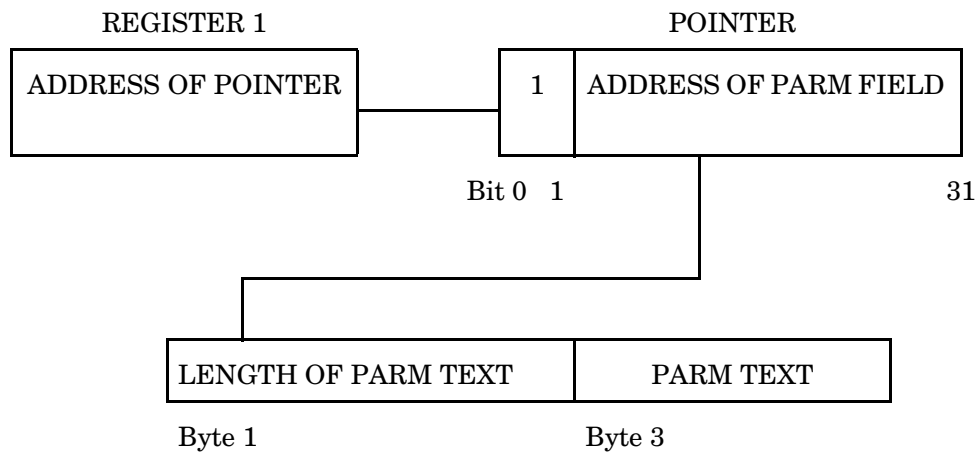
The PARMEXIT facility provides a means of dynamically modifying PARM values based on conditions at execution time, and requires a user-written program/routine.

The PARMEXIT routine must follow standard IBM linkage conventions. When the routine is entered, register 13 contains the address of a save area that can be used, register 14 contains the address that must be returned to, and register 15 contains the entry point address to the exit. The routine must save registers 2 through 14, and must restore them before control is returned to SyncSort. No parameter list is passed to the routine.

The routine must pass a return code in register 15: if a PARM is being passed to SyncSort, the return code must be 0; otherwise, 4. If the routine is passing a PARM field to SyncSort, a standard parameter list must be created, and the address of a pointer to the PARM field must be loaded into register 1. The four-byte pointer must contain a 1 in the high-order bit and the address of the PARM field in the remaining low-order 31 bits. The first two bytes of

the PARM field contain the length of the PARM field (excluding the two-byte length field); the remainder contains text of the same parameters that might otherwise be coded in the PARM field of a SyncSort EXEC statement. Leading and trailing blanks are permitted in the text; embedded blanks are not. The PARM field passed back to SyncSort cannot exceed 256 bytes.

Below is an illustration of the format of the PARMEXIT parameter list.



The routine must be assembled, and should be link-edited into the main SyncSort load module library with the load module name of \$ORTPARM. The name \$ORTPARM may be changed with the PARMNME option.

The parameters passed through the PARMEXIT facility take precedence over the PARM field of the EXEC statement and the parameter list for a program-initiated sort. Parameters presented by means of the \$ORTPARM DD statement take precedence over PARMEXIT PARM values.

The PARMEXIT and PARMTABLE facilities are mutually exclusive. If both are specified, only PARMTABLE will be processed.

The PARMNME Option

$$\text{PARMNME} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \underline{\$ORTPARM} \end{array} \right\}$$

The PARMNME option allows you to alter the PARMEXIT or PARMTABLE load module name. The default name is \$ORTPARM.

The PARMTBLE Option

PARMTBLE

The PARMTBLE option controls whether or not the PARMTABLE facility can be used. To enable the PARMTABLE facility, specify PARMTBLE. The default is not to specify PARMTBLE, and consequently to disable this facility.

The PARMTABLE facility allows you to alter SyncSort PARM values depending on the jobname, stepname, and/or procstepname of a sorting application.

To use the PARMTABLE facility, you must construct a load module containing one entry for each jobname/stepname/procstepname combination for which parameters will be specified. However, if you are specifying the same value for all sorts with the same procstepname (no matter what the jobname or stepname) or for all steps that have the same jobname, a single entry is used.

The load module must be a variable length block of variable length records, and each entry must be a variable length record. An entry is composed of four fields. Each of the first three fields is eight bytes long and may contain, respectively, a jobname, a stepname, and a procstepname. The names must be left-justified and initialized with blanks. The fourth field of an entry may have a length from 0 to 256 bytes, and must contain only information that is valid in the PARM field of the sort EXEC job control statement.

Whenever the PARMTABLE facility is used, a comparison is made between the jobname, stepname, and procstepname of the task that is being executed (first three fields of the TIOT) and the first three fields of each entry in the load module table. If a match is found, the PARM data of the matching entry is used in the sort.

A blank (X'40') in the first byte of any of the first three fields of an entry causes that field to be considered equal to the corresponding field in the TIOT. If the first byte of the first field contains X'00', that entry is skipped.

The table must be assembled and should be link-edited into the main SyncSort load module library with the load module name of \$ORTPARM. The name \$ORTPARM can be changed with the PARMNME option.

Below is a sample macro definition for constructing a PARM table.

The sample assembly establishes a core value for STEP2 of JOB1, a DYNALLOC parameter for the jobs with the procstepname of SMLSORT, and a BMSG parameter for jobs with a jobstepname of TESTJOB.

The parameters passed through the PARMTABLE facility take precedence over the PARM field of the EXEC statement and the parameter list for a program-initiated sort. Parameters presented by means of the \$ORTPARM DD statement take precedence over PARMTABLE PARM values.

The PARMTABLE and PARMEXIT facilities have similar functions, but the two facilities are mutually exclusive. Because of this, only PARMTABLE will be processed if both are specified.

The PGMIEAL Option

The PGMIEAL option specifies this parameter if program-initiated sorts are to use the z/OS IEALIMIT facility when sufficient memory is otherwise unavailable. The default is not to use the IEALIMIT facility in these cases. The IEALIMIT facility is automatically used in JCL sorts when necessary.

The PGMRC16 Option

PGMRC16

The PGMRC16 option controls whether a user abend 16 or a return code of 16 will be issued when an unsuccessful sort is program-initiated.

If PGMRC16 is specified and an error is detected, SyncSort will issue a return code of 16 instead of a user abend 16.

If PGMRC16 is not specified and an error is detected in a program-initiated sort, SyncSort's response is determined by the setting of the RC16 option.

JCL-initiated sorts are always handled according to the setting of the RC16 option.

The PREFIX Option

$\text{PREFIX} = \left\{ \begin{array}{l} n_1 \dots n_{27} \\ \underline{\text{TDS}} \end{array} \right\}$
--

The PREFIX option allows you to change the prefixes of the DSNAMES that MAXSORT generates for the intermediate output data sets.

The last character specified must always be a period, and a maximum of 27 other characters may be coded unless the TIMESTMP option is also specified. If TIMESTMP is specified, a maximum of 18 characters plus the final period may be used. TDS. is the default name.

The PREFIX option can be overridden by the BKPTDSN PARM at execution time.

The PRINT121 Option

PRINT121

The PRINT121 option controls whether or not SyncSort is to alter its DCB values for the output message data set (SYSOUT) during JCL-initiated sorts. If you specify PRINT121, the DCB values will be changed as in the figure below.

From:

DCB=(LRECL=125,BLKSIZE=882,RECFM=VBA)

To:

DCB=(LRECL=121,BLKSIZE=121,RECFM=FA)

Figure 8. DCB Changes When PRINT121 Specified

The default is not to specify PRINT121, and so not to alter the DCB values for the output message data set. The DCB settings cannot be changed during program-invoked sorts.

The RC16 Option

$$\left\{ \begin{array}{l} \text{RC16} = \left\{ \begin{array}{l} \text{ABE} \\ \text{NOABE} \end{array} \right\} \\ \text{NORC16} \end{array} \right\}$$

The RC16 option controls whether a user abend 16 or a return code of 16 will be issued for an unsuccessful sort.

If you wish to handle abends differently for JCL and program-initiated sorts, set the RC16 option for JCL-initiated sorts and set the PGMRC16 option for program-initiated sorts.

If you specify RC16=ABE, SyncSort will issue user abend 16 in the case of an unsuccessful sort. User abend 16 does not produce a dump.

If you specify RC16=NOABE (or NORC16), SyncSort will issue a return code of 16 when a sort is unsuccessful.

The RELEASE Option

$$\text{RELEASE} = \left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\}$$

The RELEASE option controls whether or not excess space on the SORTWK data sets will be released at the end of the input phase during JCL-initiated sorts where DISP=OLD is not specified on the SORTWK DD statement.

If RELEASE=ON is specified, SyncSort will release the space. This is the default. If RELEASE=OFF is specified, the excess space will not be released.

During a MAXSORT, excess space will be released only if SORTWK space exceeds the MAXWKSP value, and it will be released before the input phase begins.

Note that the RELSAMT option directly affects this option. If the RELSAMT default has not been changed, RELEASE=ON will not take effect unless at least 20 million bytes are sorted.

The RELSAMT Option

$$\text{RELSAMT} = \left\{ \begin{array}{c} \text{nnnnnnnn} \\ \underline{20000000} \end{array} \right\}$$

The RELSAMT option is used to control the release of excess space on the SORTWK data sets. The release is dependent on the amount of data sorted.

Specify an amount of data (nnnnnnnn) in bytes that must be sorted in order for excess SORTWK space to be released. The default is 20 million bytes of data.

The RESET Option

RESET

The RESET option allows you to prevent VSAM from treating an output file as a DISP=MOD data set if the output VSAM file was created using the REUSE option.

To prevent this type of VSAM processing, specify RESET. If REUSE was not specified when the data set was created, the RESET specification will be ignored and the data set will have output performed to it in a DISP=MOD fashion.

The default is to treat all VSAM output files as DISP=MOD data sets by not specifying RESET.

The RLSOUT Option

$$\left\{ \begin{array}{l} \text{RLSOUT} \\ \underline{\text{NORLSOUT}} \end{array} \right\}$$

The RLSOUT option allows you to control the release of excess disk space on output data set(s) at the end of a sort execution if the data sets were specified with DISP=NEW.

To release the excess space, specify RLSOUT.

The default is not to release the excess space by specifying NORLSOUT.

The ROUTCDE Option

$$\text{ROUTCDE} = \left\{ \begin{array}{l} (n_1, \dots, n_{16}) \\ \underline{0} \end{array} \right\}$$

The ROUTCDE option allows you to specify up to 16 route codes (n) for SyncSort messages to the console.

The default is 0.

The RSRVJI Option

$$\text{RSRVTI} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \text{nK} \\ \underline{0} \end{array} \right\}$$

The RSRVJI option is used to reserve virtual storage below the 16-megabyte line for system or exit requirements during a JCL-initiated sort.

This option will be in effect if the execution time parameter CORE is not passed, or if CORE is set to MAX when it is passed. Information about execution time parameters can be found in the *SyncSort for z/OS Programmer's Guide*.

The amount of storage to be reserved may be specified in bytes (nnnnnnnnn) or kilobytes (nK).

The default is 0.

The RSRVTI Option

$$\text{RSRVTI} = \left\{ \begin{array}{l} \text{nnnnnnnnn} \\ \text{nK} \\ \underline{44\text{K}} \end{array} \right\}$$

The RSRVTI option is used to reserve virtual storage below the 16-megabyte line for system or program requirements during a program-initiated sort.

This option will be in effect if the execution time parameter CORE is not passed, or if CORE is set to MAX when it is passed. Information about execution time parameters can be found in the *SyncSort for z/OS Programmer's Guide*.

The amount of storage to be reserved may be specified in bytes (nnnnnnnnn) or kilobytes (nK).

The default is 44K.

The SDB Option

$$\text{SDB} = \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{DISKONLY} \\ \text{TAPEONLY} \\ \text{LARGE} \\ \text{SMALL} \\ \underline{\text{INPUT}} \\ \text{LARGEONLY} \\ \text{INPUTONLY} \end{array} \right\}$$

The SDB option specifies whether system-determined blocksize should be used to select an optimal blocksize for SORTOUT, SORTOFxx, SORTOFx, and SORTXSUM data sets when none is provided. This parameter will automatically provide a blocksize that will most efficiently utilize the space on the output device.

SDB=ON enables the use of system-determined blocksize for both tape and new or previously allocated but unopened DASD output data sets except in the following conditions:

- A blocksize is found in the JCL DCB BLKSIZE specification or, in the case of a DISP=MOD tape data set, it is derived from an available tape label.
- The output file is a VSAM data set.

If the output data set is on DASD, the blocksize selected will be based upon the RECFM and LRECL, either specifically provided or determined from the usual analysis of SORTIN or RECORD statement attributes. For example, the blocksize selected for a blocked output data set assigned to a 3380 or 3390 DASD device will represent a size as close to half-track blocking as possible.

If the output file is a tape data set, the blocksize will be determined from the RECFM and LRECL in conjunction with the following rules:

- RECFM of F or FS: BLKSIZE=LRECL
- RECFM of FB or FBS and LABEL type is not AL: BLKSIZE=highest multiple of LRECL that is less than or equal to 32760.
- RECFM of FB and LABEL type is AL: BLKSIZE=highest multiple of LRECL that is less than or equal to 2048.
- RECFM of V, VS, D: BLKSIZE=LRECL +4
- RECFM of VB, VBS: BLKSIZE=32760
- RECFM of DB: BLKSIZE=2048

If SDB=OFF is specified, SyncSort will not use system-determined blocksize. The blocksize, if unavailable, will be determined from SORTIN if the SORTIN and output data set LRECLs are the same, otherwise SyncSort will select an appropriate blocksize.

If SDB=DISKONLY is specified, SyncSort will use system-determined blocksize only for disk output data sets.

If SDB=TAPEONLY is specified, SyncSort will use system-determined blocksize only for TAPE output data sets.

SDB=LARGE enables the use of system-determined blocksize for both tape and DASD output data sets, as with SDB=ON. Additionally, under Version 2 Release 10 of OS/390 or under z/OS, SDB=LARGE enables selection of a system-determined blocksize greater than 32760 for eligible tape output data sets if not restricted by the system BLKSZLIM value.

SDB=SMALL has the same meaning as SDB=ON.

SDB=INPUT enables the use of system-determined blocksize for both tape and DASD output data sets, as with SDB=ON. Additionally, under Version 2 Release 10 of OS/390 or under z/OS, if an input tape data set has a blocksize greater than 32760, SDB=INPUT enables selection of a system-determined blocksize greater than 32760 for eligible tape output data sets if not restricted by the system BLKSZLIM value. SDB=INPUT is the default.

SDB=LARGEONLY enables the use of system-determined blocksize for tape output data sets only, as with SDB=TAPEONLY. Additionally, under Version 2 Release 10 of OS/390 or under z/OS, SDB=LARGEONLY enables selection of a system-determined blocksize greater than 32760 for eligible tape output data sets if not restricted by the system BLKSZLIM value.

SDB=INPUTONLY enables the use of system-determined blocksize for tape output data sets only, as with SDB=TAPEONLY. Additionally, under Version 2 Release 10 of OS/390 or under z/OS, if an input tape data set has a blocksize greater than 32760, SDB=INPUTONLY enables selection of a system-determined blocksize greater than 32760 for eligible tape output data sets if not restricted by the system BLKSZLIM value.

The SDBGENR Option

SDBGENR =	$\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{DISKONLY} \\ \text{TAPEONLY} \\ \text{LARGE} \\ \text{SMALL} \\ \text{INPUT} \\ \text{LARGEONLY} \\ \text{INPUTONLY} \end{array} \right\}$
-----------	---

The SDBGENR option specifies whether system-determined blocksize should be used to set an optimal blocksize for SYSUT2 data sets in BetterGener applications when no blocksize is provided. This parameter will automatically provide a blocksize that will most efficiently utilize the space on the output device.

See the SDB option for a description of the parameters, substituting SYSUT1 for SORTIN and SYSUT2 for SORTOUT in the description.

The SECOND Option

SECOND =	$\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$
----------	---

The SECOND option allows you to instruct SyncSort to automatically obtain secondary allocation on SORTWK data sets even if secondary allocation has not been specified in the JCL SPACE parameters. The size of the secondary allocation can be controlled with the WKSPCE option.

To allow automatic secondary allocation, specify SECOND=ON. This is the default.

If SECOND=ON during a MAXSORT, secondary allocation will be obtained before any sorting begins.

To prohibit secondary allocation unless secondary allocation has been requested in the JCL SPACE parameters, specify SECOND=OFF.

The SMF Option

$$\text{SMF} = \left(\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} \left\{ \begin{array}{c} ,\text{nnn} \\ ,\underline{208} \end{array} \right\} \left\{ \begin{array}{c} \underline{S} \\ ,\underline{P} \\ ,\underline{D} \end{array} \right\} \right)$$

The SMF option controls the SyncSort SMF facility. This facility allows SyncSort to generate a user SMF record at the end of each SyncSort execution to describe how SyncSort ran. Information such as how much data was sorted, how much virtual storage was used, and how much disk space was allocated is collected. This data can be used to improve the efficiency of subsequent runs of the same sort application.

Specify SMF=ON if you want SyncSort SMF records to be generated. The default is SMF=OFF, which means that no SyncSort SMF records will be generated.

Use nnn to give an SMF record type number to the SyncSort SMF record. Specify a decimal number from 128 to 255. If you do not specify a number, 208 will be used.

If S (the default) is specified, SyncSort will issue the SMFWTM SVC to write the SMF records to the system SMF data set.

If P is specified, SyncSort will write to the data set defined by the \$ORTSTAT DD statement, which must be included in the JCL stream of every sort for which SMF records are to be obtained.

If D is specified, SyncSort will attempt to write to the data set normally defined by a \$ORTSTAT DD statement. If a \$ORTSTAT DD statement is not present, SyncSort will dynamically allocate that DD statement, using the data set name SYS1.SYNCSMF. A data set named SYS1.SYNCSMF must have been previously allocated and catalogued. SyncSort SMF records will then be written to SYS1.SYNCSMF.

The STATREC option allows you to tell SyncSort to generate a formatted report from its SMF records. See “The STATREC Option” on page 5.41 for complete details.

The SOPAD Option

$$\text{SOPAD} = \left\{ \begin{array}{c} \text{RC0} \\ \text{RC4} \\ \text{RC16} \end{array} \right\}$$

The SOPAD option specifies the action to be taken if the LRECL defined in the JCL for a non-OUTFIL SORTOUT is larger than the SORTIN/SORTINnn LRECL or the internally processed record length when the SORTIN/SORTINnn LRECL is modified by features.

- RC0** The delivered default specifies that SyncSort should issue a WER462I, pad fixed length output records with binary zeros, and issue a return code of zero.
- RC4** Specifies that SyncSort should issue a WER462I and pad fixed length output records with binary zeros. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- RC16** Instructs SyncSort to issue a WER462A message and to terminate processing with a return code of 16.

The SOPADGN Option

$$\text{SOPADGN} = \left\{ \begin{array}{c} \text{RC0} \\ \text{RC4} \\ \text{IEB} \end{array} \right\}$$

The SOPADGN option specifies the action to be taken in BetterGener applications if the LRECL defined in the JCL for SYSUT2 is larger than the SYSUT1 LRECL or the internally processed record length when the SYSUT1 LRECL is modified by features.

- RC0** The delivered default specifies that BetterGener should issue a WER462I, pad fixed length output records with binary zeros, and issue a return code of zero.
- RC4** Specifies that BetterGener should issue a WER462I and pad fixed length output records with binary zeros. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- IEB** Specifies that BetterGener should transfer control to IEBGENER when the SYSUT2 LRECL is larger than the SYSUT1 LRECL.

The SORTIME Option

$$\text{SORTIME} = \left(\begin{array}{c} n \\ n, \text{OC} \\ \underline{120}, \text{OC} \\ \underline{1440} \end{array} \right)$$

The SORTIME option is used to specify the minimum amount of elapsed time to be consumed by MAXSORT before automatic termination takes place.

Specify the amount of time (n) that should elapse before automatic termination. MAXSORT will take action at the next breakpoint after this amount of time elapses.

Specify OC for operator communication at the breakpoint so that the operator can decide whether to terminate the sort at that breakpoint or to continue until the next breakpoint. If no operator communication is specified, automatic termination will take place at that breakpoint.

The default if you do not specify OC (operator communication) is 1440 minutes or 24 hours. If you specify only OC, the default is 120 minutes or 2 hours. The delivered default is 1440.

The SOTRN Option

$$\text{SOTRN} = \left\{ \begin{array}{c} \text{RC0} \\ \text{RC4} \\ \text{RC16} \end{array} \right\}$$

The SOTRN option specifies the action to be taken if the LRECL defined in the JCL for a non-OUTFIL SORTOUT is smaller than the SORTIN/SORTINnn LRECL or the internally processed record length when the SORTIN/SORTINnn LRECL is modified by features.

- RC0** The delivered default specifies that SyncSort should issue a WER462I, truncate the output records, and issue a return code of zero.
- RC4** Specifies that SyncSort should issue a WER462I and truncate the output records. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- RC16** Specifies that SyncSort should issue a WER462A message and terminate processing with a return code of 16.

The SOTRNGN Option

$$\text{SOTRNGN} = \left\{ \begin{array}{l} \text{RC0} \\ \text{RC4} \\ \text{IEB} \end{array} \right\}$$

The SOTRNGN option specifies the action to be taken in BetterGener applications if the LRECL defined in the JCL for SYSUT2 is smaller than the SYSUT1 LRECL or the internally processed record length when the SYSUT1 LRECL is modified by features.

- RC0** The delivered default specifies that BetterGener should issue a WER462I, truncate the output records, and issue a return code of zero.
- RC4** Specifies that BetterGener should issue a WER462I and truncate the output records. A return code of 4 will be issued if not overridden by a higher return code set for another reason.
- IEB** Specifies that BetterGener should transfer control to IEBGENER when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL.

The STATREC Option

$$\text{STATREC} = \left\{ \begin{array}{l} \text{SYNCSMF} \\ \text{SYNCSTAT} \end{array} \right\}$$

The STATREC option allows you to generate a formatted report based on the data collected for the SyncSort SMF record.

The format of SyncSort's SMF record is laid out in member SMFDSECT. For information about the location of SMFDSECT if you are installing without SMP, see "SMFDSECT Location" on page 3.15. If you are installing with SMP, see "SMFDSECT Location" on page 4.21.

Specify SYNCSMF (default) if you want SyncSort to write its statistical data only to an SMF data set.

Specify SYNCSTAT if you want SyncSort to generate a formatted report of the statistical data.

If SYNCSTAT is specified, a DD statement must be provided in the JCL in the following format:

```
//STATOUT DD SYSOUT=*
```

The module that produces the listing (SYNCSTAT) is always installed when SyncSort is installed.

The SMF option controls the following:

- Whether or not a SyncSort SMF record is written
- The type number of the SyncSort SMF record
- Where the SyncSort SMF record is written.

For more information, see “The SMF Option” on page 5.38.

The following rules will be helpful when setting the SMF and STATREC options:

- If SMF=ON and STATREC=SYNCSTAT, the SyncSort SMF data will be written to the SMF data set, and the SYNCSTAT report will also be produced.
- If SMF=ON and STATREC=SYNCSMF, the SyncSort SMF data will be written to the SMF data set, but no SYNCSTAT report will be produced.
- If SMF=OFF and STATREC=SYNCSTAT, no SyncSort SMF data will be written to the SMF data set, but a SYNCSTAT report will be produced based on the collected data.
- If SMF=OFF and STATREC=SYNCSMF, no SyncSort SMF data will be written to the SMF data set, and no SYNCSTAT report will be produced.

The SUMOVFL Option

$$\text{SUMOVFL} = \left\{ \begin{array}{l} \text{RC0} \\ \text{RC4} \\ \text{RC16} \end{array} \right\}$$

The SUMOVFL option specifies the action to be taken if a summary field overflows or underflows during SUM processing.

RC0 The delivered default instructs SyncSort to issue a WER049I warning message and continue processing. A return code of 0 will be returned if not overridden by a higher return code set for another reason. The WER049I will only be issued on the first occurrence of the overflow or underflow.

RC4 Instructs SyncSort to issue a WER049I warning message and continue processing. A return code of 4 will be issued if not overridden by a higher return code set for another reason. The WER049I will only be issued on the first occurrence of the overflow or underflow.

RC16 Instructs SyncSort to issue a WER049A message and to terminate processing with a return code of 16.

The SVC Option

$$\text{SVC} = \left(\begin{array}{l} \text{nnn} \\ 109, \text{mm} \\ \underline{0}, \underline{0} \end{array} \right)$$

The SVC option is used to assign an SVC number to SyncSort. An SVC is needed for SyncSort's high-performance access methods, optimized DASD performance when using cache devices, and for the key data set facility. It is also used to access the SMFWTM SVC when writing records to the system SMF data set.

The SVC number specified in this option must be valid on your system, and it must be the SVC specified during SyncSort installation. Failure to provide a valid SVC number for SyncSort will cause a critical MNOTE to be issued when you are assembling the SYNCMAC macro, and you will be unable to install.

You may designate a value from 200 to 255 (nnn) for the SVC number. Be careful to select an SVC number not already in use on your system.

If you use SVC 109, you must also specify a valid router code (mm) that is not currently defined. The router code may have a value of 17 or 200 to 255.

Use the following guidelines when specifying an SVC load module name:

- If SVC=nnn is coded, assign a module name of IGC00nna where the "a" in "nna" corresponds to the last digit of the SVC number with an X'C' in the zone portion of the byte. This is the standard naming convention for a type-3 SVC. For example, if SVC=245 is specified, the load module name would be IGC0024E.

When assigning the SVC an IGC-type module name, a corresponding type-3 SVC slot must be defined via the SVCTABLE SYSGEN macro.

- If SVC=(109,mm) is coded, assign a module name of IGX000mm.
- If you use a three-digit router code (mmm), assign a module name of IGX00mmm.

In all cases, insure that the load module name you assign does not already exist on your system. If you are converting from a previous SyncSort release, use a different SVC number for SyncSort than the one you used for the previous release to facilitate testing.

Although the delivered default is 0 for both the SVC number and the router code, you cannot install SyncSort unless you substitute a valid SVC number for the delivered default.

The TAPENME Option

$\text{TAPENME} = \left\{ \begin{array}{c} \text{nnnnnnnnn} \\ \text{TAPE} \end{array} \right\}$
--

The TAPENME option sets the generic name of the tape unit (nnnnnnnnn) that MAXSORT can use for the dynamic allocation of tapes. The default is TAPE.

The TAPENME option takes effect only when the DYNATAPE option is also specified.

The TIMESTMP Option

TIMESTMP

The TIMESTMP option will generate a time stamp as part of each data set name of the intermediate output data sets of a MAXSORT.

The time stamp will appear as an index in the data set names in the form "Ddddhhmm." with ddd the Julian day, hh the hour, and mm the minute.

To generate a time stamp, specify TIMESTMP. The default is not to generate a time stamp by not specifying TIMESTMP.

The TITLE Option

$\text{TITLE} = 'c_1 \dots c_{84}'$

The TITLE option allows a user-supplied installation heading of up to 84 characters (c) to be written on the second line of each SyncSort heading. The default is blanks.

Note: A single ampersand character (&) is not accepted by IEV90 and ASMA90 assemblers. If you want an ampersand in the TITLE option, use two ampersands (&&).

The UNINTDS Option

$$\text{UNINTDS} = \left\{ \begin{array}{c} \text{YES} \\ \underline{\text{NO}} \end{array} \right\}$$

The UNINTDS option indicates how SyncSort should process a non-VSAM uninitialized DASD SORTIN or SORTINnn data set in a non-SMS environment. An uninitialized data set is one that has been created but never successfully opened and closed for output. In an SMS environment, uninitialized data sets are always processed as valid empty files.

UNINTDS=YES indicates that an uninitialized data set should be processed as an empty file. If an uninitialized multi-volume data set has the DS1IND80 and DS1IND02 flags off in the format-1 DSCB of the first volume and the number of data extents is non-zero, SyncSort will open the data set for output to set an end-of-file mark before the data set is used for input.

UNINTDS=NO indicates that SyncSort should terminate with a WER400A critical message if an uninitialized data set is provided as input on SORTIN or SORTINnn.

The UNINTGN Option

$$\text{UNINTGN} = \left\{ \begin{array}{c} \text{YES} \\ \underline{\text{NO}} \end{array} \right\}$$

The UNINTGN option indicates how SyncSort's BetterGener facility should process a non-VSAM uninitialized DASD SYSUT1 data set in a non-SMS environment. An uninitialized data set is one that has been created but never successfully opened and closed for output. In an SMS environment, uninitialized data sets are always processed as valid empty files.

UNINTGN=YES indicates that an uninitialized data set should be processed as an empty file. If an uninitialized multi-volume data set has the DS1IND80 and DS1IND02 flags off in the format-1 DSCB of the first volume and the number of data extents is non-zero, BetterGener will open the data set for output to set an end-of-file mark before the data set is used for input.

UNINTGN=NO indicates that BetterGener should terminate with a WER400A critical message if an uninitialized data set is provided as input on SYSUT1.

IEBGENER will treat an uninitialized data set as an empty data set. If you want to replicate this processing in BetterGener, set UNINTGN=YES. SyncSort (non-BetterGener) processing of uninitialized data sets can be separately controlled by the UNINTDS installation option.

The VLTEST Option

$$\text{VLTEST} = \left(\begin{bmatrix} n \\ 1 \\ - \end{bmatrix} \begin{bmatrix} ,\text{ON} \\ ,\text{OFF} \\ ,\text{OFF4} \end{bmatrix} \right)$$

The VLTEST option allows you to do the following when variable length records are processed:

- Choose the type of record length validity testing to be performed.
- Choose whether or not to verify the correct sequence of segments in variable length spanned records.

Record length validity testing may be performed in all types of applications: sort, merge, copy, and BetterGener. Segment sequence checking may only be done during sort and merge applications.

The n value in the first subparameter specifies the type of record length validity testing to be performed, and the chart below gives the values of n that can be specified.

0*	No record length validity testing of variable length records.
1	If any input record does not contain all SORT/MERGE control fields, terminate. This is the default.
2*	If any input record is longer than the maximum LRECL or l2 value, terminate.
3	If either or both of the conditions in tests 1 and 2 are satisfied, terminate.
4*	If any input record is longer than the output LRECL or l3 value, terminate.
5	If either or both of the conditions in test 1 or 4 are satisfied, terminate.
6*	If any input record is longer than the maximum input LRECL or l2 value, or longer than the output LRECL or both, terminate.
7	If any of the conditions in test 1, 2, or 4 are satisfied, terminate.
<i>* These values force the use of CMP=CLC for variable length input.</i>	

The second subparameter allows you to specify whether or not SyncSort should verify that the sequence of segments is correct in each variable length spanned record during sort and merge applications. ON is the delivered default and signals that the segment sequence

should be verified. If OFF is selected, all illogical record segments encountered in the input file will be processed as individual records. If OFF4 is selected, the processing described for OFF will occur, but in addition if an illogical segment is found, message WER464I will be produced and a return code of 4 will be returned if not overridden by a higher return code set for another reason.

The second subparameter does not apply during copy applications.

One of the following error messages will be issued if segment sequence is ON and an illegal condition is detected during a validity test: WER027A, WER160A, WER167A, or WER182A.

The VLTESTI Option

$$\text{VLTESTI} = \left\{ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \right\}$$

The VLTESTI option specifies to SyncSort how to process variable length records that do not contain all specified INCLUDE or OMIT fields. VLTESTI applies to both regular and OUTFIL INCLUDE/OMIT processing.

The delivered default of 0 instructs SyncSort to terminate if a record does not completely contain all INCLUDE or OMIT fields. A WER250A critical error message is generated to indicate this condition.

When VLTESTI=1 is specified, a record that does not completely contain all INCLUDE/OMIT fields is treated as having failed the comparison. SyncSort will omit the record if INCLUDE is being used or include the record if OMIT has been specified.

When VLTESTI=2 is specified, SyncSort will treat comparisons to fields not completely contained within the record as false and decide a record's status for inclusion or omission from fields that are available. If all fields are not present, the record will be processed as having failed the comparison. SyncSort will omit the record if INCLUDE is being used or include the record if OMIT has been specified.

The VSAMEMT Option

$$\text{VSAMEMT} = \left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\}$$

The VSAMEMT option specifies the processing of empty VSAM data sets provided as input to a sort, merge, or copy.

If you specify VSAMEMT=YES, an empty VSAM data set will be processed as a legitimate data set containing 0 records, and SyncSort will end with a return code of 0.

The delivered default, VSAMEMT=NO, instructs SyncSort to terminate with a WER254A critical error if an empty VSAM data set is specified for input.

The VSCORE Option

$$\text{VSCORE} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \text{nK} \\ \text{nM} \\ \underline{1\text{M}} \end{array} \right\}$$

The VSCORE option allows you to set the maximum amount of virtual storage below the 16-megabyte line that SyncSort can use for its working set.

The VSCORE limit cannot be overridden by the execution time parameter CORE. However, it can be overridden by the execution time parameter VSCORE.

Use nnnnnnnn to give the storage limit in bytes, nK to give the storage limit (n) in kilobytes, and nM to give the storage limit (n) in megabytes. The delivered default is 1M.

The VSCORET Option

$$\text{VSCORET} = \left\{ \begin{array}{l} \text{nnnnnnnn} \\ \text{nK} \\ \text{nM} \\ \underline{6\text{M}} \end{array} \right\}$$

The VSCORET option allows you to set the maximum amount of virtual storage below and above the 16-megabyte line that SyncSort can use for its working set when SyncSort's Dynamic Storage Management (DSM) facility is inactive.

When DSM is active, VSCORET is treated as an initial recommendation.

Without DSM, SyncSort cannot dynamically adjust its use of virtual storage in response to system activity. For the best performance of SyncSort and your system, DSM is strongly recommended. If you choose to disable DSM, however, you should determine the VSCORET value based on factors such as the amount of central storage on your system, sort filesize, page activity, and the number of sorts which would typically run concurrently. **Important:**

Do not disable DSM without first contacting SyncSort for z/OS Product Services for advice on setting this option.

The VSCORET limit cannot be overridden by the execution time parameter CORE. However, it can be overridden by the execution time parameter VSCORET.

The VSCORET value cannot be less than the value set in the VSCORE option.

Use nnnnnnnn to give the storage limit in bytes, nK to give the storage limit (n) in kilobytes, and nM to give the storage limit (n) in megabytes. The delivered default is 6M.

The WKSPCE Option

$\text{WKSPCE} = (\left\{ \begin{array}{c} \text{ccccccc} \\ \underline{10} \end{array} \right\} \left\{ \begin{array}{c} \text{,ttttttt} \\ \underline{,200} \end{array} \right\} \left\{ \begin{array}{c} \text{,bbbbbbb} \\ \underline{,400} \end{array} \right\} [\text{,ppp}\%])$
--

The WKSPCE option allows you to specify the amount of space for each secondary extent that SyncSort will obtain for each SORTWK disk data set if no secondary allocation is specified in the JCL SPACE parameter. However, no space will be obtained unless the SECOND option is set to ON.

Use ccccccc to give an absolute quantity in cylinders. The default is 10 cylinders, and only sorts with primary allocations specified in cylinders are affected.

Use ttttttt to give an absolute quantity in *tracks*. The default is 200 tracks, and only sorts with primary allocations specified in tracks are affected.

Use bbbbbbb to give an absolute quantity in *blocks*. The default is 400 blocks, and only sorts with primary allocations specified in blocks are affected.

When present, the ppp% subparameter is used to override the defaults for cylinders, tracks, and blocks. It sets the percentage (ppp) of the primary allocation that is to be allocated as space for each secondary extent.

The following rules are important when using ppp%:

- The percent sign (%) must be specified. For example, 80% is a valid value for ppp%.
- It is illegal to specify ppp% unless at least one of the other subparameters is omitted. For example, to request cylinder allocations of 15 secondary cylinders, track allocations of 200 tracks, and block allocations of an additional 20% in work space, specify WKSPCE=(15,200,,20%). Note that, even though 200 is the default for ttttttt, it must be specified here because all defaults are overridden when ppp% is present. A percentage of the primary allocation is used instead.

- The ppp% parameter itself has no default value. When ppp% is omitted, any other omitted subparameter receives the default value indicated and no percentage is taken.

The ZDPRINT Option

$$\text{ZDPRINT} = \left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}$$

The ZDPRINT option specifies if positive ZD summation results are to be converted to printable numbers. This option controls the handling of ZD fields specified on a SUM control statement.

ZDPRINT determines whether the sign byte of a positive summarized ZD field will be converted to a printable format. More precisely, the option specifies whether the zone of the last digit should be changed from a hexadecimal C to a hexadecimal F.

ZDPRINT=YES, the default, enables the conversion of positive ZD summation results to printable numbers.

ZDPRINT=NO will prevent the conversion of a positive ZD summation result to a printable number.

Setting Tape Sort Defaults

Tape Sort allows you to use tape SORTWK data sets.

Most of the Tape Sort options are set only with a zap. However, note the following exceptions:

- LISTJ, LISTT, and SVC are set only in the SYNCMAC macro. The SVC number must be specified; use the same number as for the disk sort.
- ALTMSG and MSGDD are set both with a zap and in the SYNCMAC macro.

All the zaps needed to change Tape Sort defaults are described in this section. These zaps must be coded in the form required for the SPZAP Service Aid Program or its equivalent.

For information on changing the defaults during installation, see “STEP 5: Choose SyncSort Options” on page 3.6 if you are installing without SMP and “STEP 11: Set and Install the SyncSort Options” on page 4.10 if you are installing with SMP.

Because setting JCLCORE, PGMCORE, and Tape Density are unique to Tape Sort, they are explained in detail below. All of the other options are explained in detail earlier in this chapter.

ALTMSG

ZAP	DESCRIPTION
NAME WERTSORT VER 002C E2E8E2D6E4E34040 REP 002C xxxxxxxxxxxxxxxx	<i>Use for non-SMP and SMP SYSOUT (Default) 8-Character DDNAME Substitute</i>

The substitute DDname must contain eight characters, and it must be left justified and padded with blanks.

ALTMSG=MSGOUT NAME WERTSORT VER 002C E2E8E2D6E4E34040 REP 002C D4E2C7D6E4E34040
--

Figure 9. ALTMSG Example for non-SMP and SMP

ALTPARM

ZAP	DESCRIPTION
NAME WERRCM WER8CM NAME WER8CM VER 0DF4 5BD6D9E3D7C1D9D4 REP 0DF4 xxxxxxxxxxxxxxxx	<i>Use for non-SMP Use for SMP \$ORTPARM (Default) 8-Character DDNAME Substitute</i>

The substitute DDname must contain eight characters, and it must be left justified and padded with blanks.

ALTMSG=\$ALTPARM NAME WERRCM WER8CM VER 0DF4 5BD6D9E3D7C1D9D4 REP 0DF4 5BC1D3E3D7C1D9D4
--

Figure 10. ALTMSG Example for Non-SMP

IOERR

ZAP	DESCRIPTION
NAME WERTSORT VER 0023 00 REP 0023 01	<i>Use for non-SMP and SMP IOERR=NOABE or NOIOERR (Default) IOERR=ABE</i>

JCLCORE

Use JCLCORE to specify the amount of virtual storage SyncSort can use for JCL-initiated sorts using Tape Sort.

Here is the zap for changing the JCLCORE default for Tape Sort:

ZAP	DESCRIPTION
NAME WERRCM WERAM1 NAME WERAM1 VER 0000 00000000 REP 0000 xxxxxxxx REP 0000 yyyyyyyy	<i>Use for non-SMP</i> <i>Use for SMP</i> <i>JCLCORE=MAX (Default)</i> <i>JCLCORE=VALUE</i> <i>JCLCORE=MAX-VALUE</i>

Use MAX (default) when SyncSort should use *all* the virtual storage available within the region for JCL-initiated sorts.

Use VALUE to specify the amount of virtual storage available within the region in bytes that SyncSort should use for JCL-initiated sorts. The amount must be given in hexadecimal.

Use MAX-VALUE to tell SyncSort to use all the virtual storage available within the region *less* a specific amount of virtual storage for JCL-initiated sorts. The amount of virtual storage must be expressed in two's complement.

NAME WERRCM WERAM1 VER 0000 00000000 REP 0000 00080000
--

Figure 11. Example for Non-SMP with JCLCORE=512K

NAME WERAM1 VER 0000 00000000 REP 0000 FFFC0000

Figure 12. Example for SMP with JCLCORE=MAX-256K

MSG

ZAP	DESCRIPTION
NAME WERRCM WERAM1 NAME WERAM1 VER 0006 20 REP 0006 00 REP 0006 01 REP 0006 10 REP 0006 11 REP 0006 02 REP 0006 22 REP 0006 12 REP 0006 21	<i>Use for non-SMP</i> <i>Use for SMP</i> <i>AP (Default)</i> <i>NO</i> <i>CC or NOFLAG</i> <i>CP</i> <i>CB or FLAG(U)</i> <i>AC</i> <i>AB or PC</i> <i>SP</i> <i>SC or FLAG(I)</i>

MSGDD

ZAP	DESCRIPTION
NAME WERTSORT VER 0024 E2E8E2D6E4E34040 REP 0024 xxxxxxxxxxxxxxxxx	<i>Use for non-SMP and SMP</i> <i>SYSOUT (Default)</i> <i>8-Character DDNAME Substitute</i>

The substitute DDname must contain eight characters, and it must be left justified and padded with blanks.

NAME WERTSORT VER 002C E2E8E2D6E4E34040 REP 002C D4E2C7D6E4E34040

Figure 13. Example for Non-SMP and SMP for MSGDD=MSGOUT

PGMCORE

Use PGMCORE to specify the amount of virtual storage SyncSort can use for program-initiated sorts using Tape Sort.

Here is the zap for changing the PGMCORE default for Tape Sort:

ZAP	DESCRIPTION
NAME WERRCM WER8CM NAME WER8CM VER 0024 00014000 REP 0024 xxxxxxxx REP 0024 00000000 REP 0024 yyyyyyyy	<i>Use for non-SMP</i> <i>Use for SMP</i> <i>PGMCORE=80K (Default)</i> <i>PGMCORE=VALUE</i> <i>PGMCORE=MAX</i> <i>PGMCORE=MAX-VALUE</i>

The default value, 80K, indicates how much virtual storage a program-initiated sort will use. Note that the default is specified in hexadecimal in the zap.

Use VALUE to specify the amount of virtual storage in bytes available within the region that SyncSort should use for program-initiated sorts. The amount must be given in hexadecimal.

Use MAX when SyncSort should use *all* the virtual storage available within the region for program-initiated sorts.

Use MAX-VALUE to tell SyncSort to use all the virtual storage available within the region *less* a specific amount of virtual storage for program-initiated sorts. The amount of virtual storage must be expressed in two's complement.

NAME WERRCM WER8CM VER 0024 00014000 REP 0024 00050000
--

Figure 14. Example for Non-SMP and PGMCORE=320K

NAME WER8CM VER 0024 00014000 REP 0024 FFFFC000

Figure 15. Example for SMP and PGMCORE=MAX-16K

RC16

ZAP	DESCRIPTION
NAME WERTSORT VER 0022 00 REP 0022 01	<i>Use for non-SMP and SMP</i> <i>RC16=NOABE or NORC16 (Default)</i> <i>RC16=ABE</i>

ROUTCDE

ZAP	DESCRIPTION
NAME WERTSORT VER 0036 0020 REP 0036 xxxx	<i>Use for non-SMP and SMP ROUTCDE=11 (Default) Desired route codes in binary/hex</i>

To find the value for xxxx, do the following:

- Begin with the binary number B'0000000000000000'.
- Turn on any of the 16 bits in the binary number corresponding to the route codes from 1 to 16 that you want to use.
- Convert the binary number to hexadecimal.
- Substitute the hexadecimal number for xxxx in the zap.

NAME WERTSORT VER 0036 0020 REP 0036 0402

Figure 16. Example for Non-SMP and SMP for ROUTCDE=6 and 15

The sixth bit and the fifteenth bit were set on in the binary number, B'0000010000000010', signaling the use of route codes 6 and 15. The resulting binary number was then converted to hexadecimal (X'0402'), and X'0402' was substituted for xxxx in the zap.

Tape Density Default

If you are using tape SORTWK data sets on dual-density drives, you should change the tape density default for Tape Sort from 1600 BPI (the default) to 6250 BPI. This allows SyncSort to determine the capacity of a reel of tape correctly.

Here is the zap for changing the tape density default for Tape Sort:

ZAP	DESCRIPTION
NAME WERTSORT VER 0038 14 REP 0038 18	<i>Use for non-SMP and SMP 1600 BPI (Default) 6250 BPI</i>

The SYNCLIST Report Program

The SYNCLIST report program is an aid to SyncSort installation and maintenance. It produces a formatted listing of the current status of SyncSort which includes:

- All generated options from SYNCMAC
- All Special Customer Zaps (SCZS) applied
- The current maintenance (TPF) level
- System environmental information
- A PARASORT configuration report and esoteric unit name table

SYNCLIST may be used to determine all the current SyncSort options prior to installing a new release of SyncSort.

SYNCLIST is automatically installed during SyncSort installation.

Job Control Language

The following example shows a sample execution of SYNCLIST:

```
//STEP1      EXEC    PGM=SYNCLIST
//STEPLIB    DD      DSN=SYNCSORT.LINKLIB,DISP=SHR
//SYSPRINT   DD      SYSOUT=*
//SYSLIB     DD      DSN=SYNCSORT.LINKLIB,DISP=SHR
//           DD      DSN=SYNCSORT.SYNCRENT,DISP=SHR
```

Figure 17. Sample JCL Stream for SYNCLIST

1. The STEPLIB DD statement instructs the system to locate SYNCLIST in the library SYNCSORT.LINKLIB. Since SYNCLIST resides in the same library as SyncSort, the STEPLIB DD statement is not required if the SyncSort library is in the LINKLIST.
2. The SYSPRINT DD statement assigns the SYNCLIST status report to the output device associated with class *.
3. The SYSLIB DD statement indicates the SyncSort library for which a SYNCLIST status report is desired. You must set up a SYSLIB that concatenates the SYNCSORT LINKLIB and SYNCRENT libraries.

PARM Values for the Systems Programmer

There are many PARM values that the applications programmer can use to control the execution of SyncSort. These parameters are described in the *SyncSort for z/OS Programmer's*

Guide. In addition, several PARMs are useful to the systems programmer. For convenience, these parameters are listed and described briefly below.

Systems programmers can enter PARM values through any of the following:

- The PARM field of the EXEC statement of a JCL-initiated sort
- The \$ORTPARM data set
- The PARMEXIT facility
- The PARMTABLE facility

Most of the parameters mentioned in this section have default settings that can be changed through the options in the SYNCMAC macro. However, in some cases, the spelling differs slightly.

Of the PARM values discussed in this section, only EMSPACE and ROUTCDE are supported when Tape Sort is used.

Maintenance PARM Values

You may need to pass the options in the following list for special circumstances or during maintenance. Complete information on these options is given earlier in this chapter unless otherwise noted in the list.

- ASR

ASR=OFF disables SyncSort's Automatic Sort Retry facility. For more information on this facility, see "Chapter 8. Debugging".

- EMSPACE

EMSPACE sets the amount of virtual storage below the 16-megabyte line that SyncSort is to reserve when it is executing. EMSPACE is specified in bytes (nnnnnnnn) or kilobytes (nK).

- MINCORE

MINCORE specifies the minimum virtual storage for SyncSort in bytes (nnnnnnnn) or kilobytes (nK). It is needed when you are executing a SyncSort job with a large number of control statements.

- ROUTCDE

ROUTCDE sets the route code that SyncSort will use to send messages to the console.

- SVC

SVC allows you to set the SyncSort SVC number along with a router code.

Performance PARM Values

The options in the following list are normally passed as PARMs for performance tuning. Complete information on these options is given earlier in this chapter unless otherwise noted in the list.

- Optimization modes

You may pass one of four PARMs to reset SyncSort's optimization mode. **BALANCE** tells SyncSort to provide the best balance of CPU time, sort elapsed time, and I/O activity. **CPU** tells SyncSort to minimize CPU time, **ELAP** tells SyncSort to minimize elapsed time, and **IO** tells SyncSort to minimize its I/O activity. For a complete explanation of these values, see "The DSM and DSMWEND Options" on page 5.8.

- BMSG

Passing this PARM causes SyncSort to print a number of optional B-type messages, which are useful for tuning purposes.

- STATREC

A PARM of STATREC=SYNCSTAT directs SyncSort to produce a formatted report of its SMF data. The delivered default (SYNCSMF) disables the generation of this report.

- VSCORE

VSCORE sets the maximum amount of virtual storage below the 16-megabyte line that SyncSort can use for its working set. Virtual storage can be specified in bytes (nnnnnnnn), kilobytes (nK), or megabytes (nM).

- VSCORET

VSCORET sets the maximum amount of virtual storage below and above the 16-megabyte line that SyncSort can use for its working set when DSM is inactive. Virtual storage can be specified in bytes (nnnnnnnn), kilobytes (nK), or megabytes (nM).

MAXSORT PARM Values

The PARMS in the following list can be passed during a MAXSORT.

- BKPTTIME

The values for BKPTTIME are the same as those for the BRKPTIM option. See "The BRKPTIM Option" on page 5.4.

- MAXMERGE

The values for MAXMERGE are the same as those for the MAXMERG option. See “The MAXMERG Option” on page 5.20.

- MINMERGE

The values for MINMERGE are the same as those for the MINMERG option. See “The MINMERG Option” on page 5.22.

- SORTTIME

The values for the SORTTIME are the same as those for the SORTIME option. See “The SORTIME Option” on page 5.40.

Changing Default Option Settings

The procedure for changing default option settings depends on whether you have installed with non-SMP or SMP. The two procedures are described in the following two sections.

Changing Default Option Settings for Non-SMP Installations

If you are not using SMP and you need to change the settings of SyncSort’s program product options, do the following:

1. Decide on the new setting for the default option(s) you wish to change by reviewing the information about them in this chapter.
2. If you are changing SyncSort options, place the new setting(s) in member SS11SVEC in your PPOPTION data set. If you are changing Tape Sort options, place the new setting(s) in members SS11TVEC and SS11TZAP. For information about setting Tape Sort options, see “Setting Tape Sort Defaults” on page 5.50.
3. Review, modify, and then submit member SS11OPCH in the sample JCL library that was created when SyncSort was installed. Any changes you need to make to the JCL are described in the comments in the jobstream.
4. If you have changed the SVC option to a new number, then rename the SVC module in accordance with the IBM SVC naming convention. Use an online facility such as ISPF/PDF to rename the SVC, which resides in the SyncSort LPALIB.
5. If you have changed the COBEXIT option from COB1 to COB2, then two additional support modules (SYNCCOB and SYNCCBIF) must be link-edited into your SyncSort LINKLIB. This link-editing may have been accomplished during the original installation process. To determine this, check your SyncSort LINKLIB for modules SYNCCOB and SYNCCBIF. If the modules are not in the LINKLIB, then review, modify, and submit the sample JCL member named SS11L370 or SS11COBL,

depending on whether you have or do not have an LE/370 environment, respectively. Any changes you need to make are described in the comments in the jobstream.

6. If you have changed the LOCALE option from the default of NONE, then an additional support module (SS11CEEM) must be link-edited into your SyncSort LINKLIB. This link-editing may have been accomplished during the original installation process. To determine this, check your SyncSort LINKLIB for module SS11CEEM. If the module is not in the LINKLIB, then review, modify and submit member SS11L370 in the sample JCL library. Any changes you need to make are described in the comments in the jobstream.
7. Update the z/OS system by refreshing LLA. In addition, if SyncSort is configured as system resident, place the modified modules into the active LPA, either by IPLing your system or by using a utility with direct update capability. If ALTSEQ, DSM, DSMWEND and SVC are not changed, then an IPL or use of a utility with direct update capability is not required. However, if you do change any of these four options in an installation configured as system resident, then you will need to use an updating utility to replace certain modules. Changing option ALTSEQ causes a change to the load modules named SS11N04, SS11X21, SS11N08 and the alias named SS11SL#5. Changing option DSM or DSMWEND causes a change to the load module named SS11N01C and the alias SS11SL#4. You need to IPL or use a utility with direct update capability to replace the changed modules.

Instructions that accompany TPF maintenance tapes may supersede the instructions for changing default option settings. You should always review all the TPF documentation you have received before using the procedure described above.

Changing Default Option Settings for SMP Installations

If you are using SMP and you need to change the settings of SyncSort's program product options, do the following:

1. Use the basic procedure described in "STEP 11: Set and Install the SyncSort Options" on page 4.10. The only difference is that you must provide a new USERMOD name and supersede the original USERMOD. For example, if your original USERMOD was named SZSS111, you might want to call the new USERMOD SZSS112. To do this, you would code the following MCS statements:

```
++USERMOD (SZSS112) .  
++VER (Z038) FMID (BSSIZ11) SUP (SZSS111) .  
.  
.  
.
```

2. If your new options do not include a new SVC number, changing the COBEXIT option from COB1 to COB2, or changing the LOCALE option from NONE, then skip to step 3.

If your new options do include changing the settings for SVC, COBEXIT, or LOCALE, then perform the applicable substeps, which relate to these three options.

- a. If your new options include a new SVC number, do the following after you APPLY the USERMOD that has your new options:

- 1) Use option 3.1 of ISPF/PDF to rename the SVC module in the SYNCSORT.R11.SYNCLPA data set. For example, if you were changing from SVC=234 to SVC=245, you would rename member IGC0023D to IGC0024E in the SYNCSORT.R11.SYNCLPA data set.

- 2) Create and submit an SMP/E job with the following UCLIN command to inform your CSI of the module renaming performed in step 1:

```
SET BDY(SYNCTGT)      /* TARGET ZONE */ .
UCLIN .
ADD LMOD(new_name) SYSLIB(SYNCLPA) RENT
      LASTUPDTYPE(ADD) LASTUPD(BSSIZ11)
++LMODIN
  ENTRY   QKQCSVC
  ORDER   QKQCSVC,QKXCIOI
  INCLUDE SYNCLIB(QKQCSVC,QKXCIOI)
++ENDLMODIN .
REP MOD(QKQCSVC) LMOD(new_name)
      LASTUPDTYPE(ADD) LASTUPD(BSSIZ11) .
REP MOD(QKXCIOI) LMOD(SS11X00,new_name)
      LASTUPDTYPE(ADD) LASTUPD(BSSIZ11) .
DEL LMOD(old_name) .
ENDUCL .
```

Assuming the example names, you would change the variable 'new_name' to IGC0024E and change the variable 'old_name' to IGC0023D.

Note: The ++LMODIN and ++ENDLMODIN statements must begin in column 1, and the ENTRY, ORDER, and INCLUDE statements must **not** begin in column 1. The other statements may begin in any column, but no statement should extend beyond column 72.

- b. If you have changed the COBEXIT option from COB1 to COB2, you will need to RECEIVE and APPLY SyncSort function BSSIC11 (for VSCOBOL II) or BSSIL11 (for COBOL/370). Keep in mind that you must have SMP/E Release 1.8.1 or higher to install either function. Choose whichever function is appropriate to your system. Both can be found on the installation tape. These functions may already have been installed during the original installation. To determine this, check your SyncSort LINKLIB for modules SYNCCOB and SYNCCBIF.

- c. If you have changed the LOCALE option from the default of NONE, you must RECEIVE and APPLY SyncSort function BSSIL11. Keep in mind that you must have SMP/E Release 1.8.1 or higher to install this function. If you are not sure whether you previously installed this function, check your SyncSort LINKLIB for module SS10CEEM.
3. Update the z/OS system by refreshing LLA. In addition, if SyncSort is configured as system resident, place the modified modules into the active LPA, either by IPLing your system or by using a utility with direct update capability.

If ALTSEQ, DSM, DSMWEND and SVC are not changed, then an IPL or use of a utility with direct update capability is not required. However, if you do change any of these four options in an installation configured as system resident, then you will need to use an updating utility to replace certain modules. Changing option ALTSEQ causes a change to the load modules named SS11N04, SS11X21, SS11N08 and the alias named SS11SL#5. Changing option DSM or DSMWEND causes a change to the load module named SS11N01C and the alias SS11SL#4. You need to IPL or use a utility with direct update capability to replace the changed modules.

Chapter 6. The BetterGener Utility Program

The BetterGener utility program offers improved performance over IEBGENER in straight copy tasks and is transparent to users when it is installed with SyncSort for z/OS.

Implementing BetterGener will provide significant savings in system resources such as elapsed time, CPU time, and channel utilization through improved I/O handling.

Once BetterGener is activated, eligible IEBGENER job streams are automatically processed by SyncSort. There is no need to replace working IEBGENER job streams with explicit requests for SyncSort's more efficient copying. BetterGener's SYNCGENR module will intercept calls to IEBGENER and determine whether the particular copy application should be processed by SyncSort. In situations supported by SyncSort, SYNCGENR will link to the sort, generating the environment of a program-initiated sort execution. In situations not supported by SyncSort, SYNCGENR will transfer control to IEBGENER.

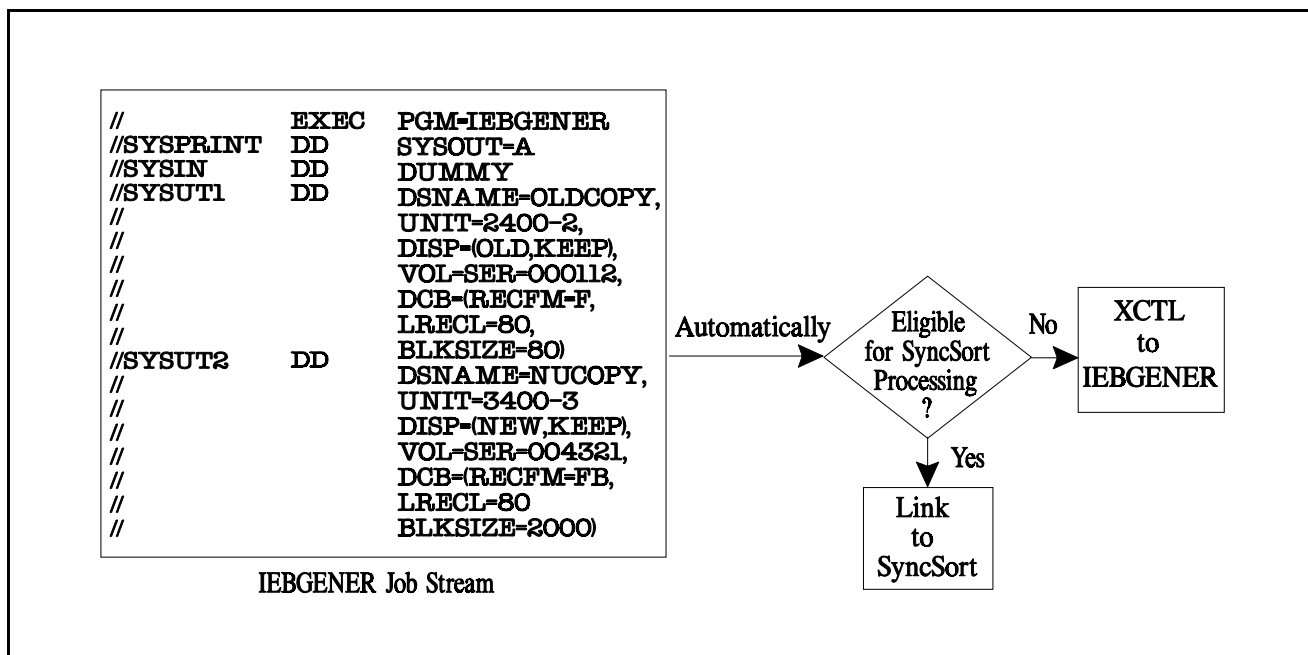


Figure 18. BetterGener Processing Flow

Eligibility for SyncSort Processing

Certain IEBGENER job streams are not eligible for SyncSort processing. For instance, IEBGENER SYSIN control statements are not supported by SyncSort. Under such circumstances, SYNCGENR will transfer control to IEBGENER.

BetterGener will also process applications not supported by IEBGENER. Automatic padding of fixed length records and truncation of fixed and variable length records will be done when necessary, based upon SYSUT2 LRECL. Incomplete SYSUT2 DCB specifications will be processed consistent with SyncSort use. This means that output data set DCB attributes will always be well-defined, which is not always true when IEBGENER is used. Support for VSAM data sets is also provided.

BetterGener may require approximately 200K more memory than IEBGENER.

Installing and Activating BetterGener

If you are installing without using SMP, the BetterGener module (SYNCGENR) is automatically installed when you install SyncSort for z/OS. To activate the module, see “Activating BetterGener” on page 3.12.

If you are installing with SMP, see “Activating BetterGener” on page 4.17.

There are three SyncSort installation options specifically designed for BetterGener:

- MSGGENR indicates where messages are to be routed.
- SDBGENR controls the use of system determined blocksize for SYSUT2.
- UNINTGN indicates how an uninitialized SYSUT1 data set is to be processed.

Using \$ORTPARM with BetterGener

Activating BetterGener provides additional "IEBGENER" editing capability by permitting the inclusion of the \$ORTPARM DD statement in IEBGENER job streams. In order for the contents of the \$ORTPARM data set to affect the copy execution, the copy must be done through SyncSort; if SYNCGENR transfers control to IEBGENER, the \$ORTPARM DD statement is ignored. Therefore, whenever SyncSort file editing is crucial to a copy application, explicit FIELDS=COPY SyncSort coding should be used. Here is an example of such a jobstream:

```
//          EXEC      PGM=IEBGENER
//SYSPRINT   DD        SYSOUT=*
//SYSUT1     DD        DSN=OLDTAPE,UNIT=TAPE,
//                      DISP=OLD,VOL=SER=000629,
//                      DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSUT2     DD        DSN=NEWDISK,UNIT=3390,DISP=(NEW,KEEP),
//                      VOL=SER=001492,SPACE=(CYL,(2,1,1)),
//                      DCB=(RECFM=FB,LRECL=80,BLKSIZE=)
//SYSIN      DD        DUMMY
//$ORTPARM   DD        *
              INREC  FIELDS=(1,8,10X,9,62)
/*
```

Figure 19. Sample Job Where SyncSort Must Be Used

In this example, \$ORTPARM acts to right-pad the first (8-byte) field of the record with 10 blanks, truncating the final 10 bytes, *if SyncSort is executed*. If, however, BetterGener routes the job back to IEBGENER, no padding or truncation is done.

BetterGener Error Messages

When BetterGener is activated, IEBGENER job streams may issue SyncSort error messages. SyncSort messages can be identified by the WER prefixed to the message number.

Chapter 7. PARASORT

PARASORT improves elapsed time performance for sorts whose input is a multi-volume tape data set and/or concatenated tape data sets. Reduced elapsed time can help critical sort applications achieve batch window goals.

The performance improvement from PARASORT is a result of processing the SORTIN input volumes in a parallel fashion. Depending upon the resources provided, elapsed time can be reduced up to 20% for 2-way input and up to 33% for 4-way input.

PARASORT requires additional tape units for the application. You will need from two to eight times the current number of tape units, depending upon resource availability and the degree of improvement desired. PARASORT automatically manages the tape units and minimizes the use of the tape drive resources by deallocating excess tape drives during initialization and releasing all the extra units at the end of the sort input phase.

The additional tape units are defined to PARASORT on up to four DDs labeled SORTPAR1, SORTPAR2, SORTPAR3 and SORTPAR4. A segment of SORTIN will be read in parallel from each of these DDs. The segmentation of SORTIN is automatic.

Creating PARASORT Esoteric Unit Names

Increased parallel input processing (up to four SORTPARn DDs) increases the elapsed time benefit. However, for optimal PARASORT performance, SyncSort must be able to read each SORTPARn input DD simultaneously with no channel contention.

Generally, the normal allocation of tape drives will not ensure sufficient channel path availability, particularly for a 4-way PARASORT (four SORTPARn DD statements). Therefore,

you may need to create **special esoteric unit names** for PARASORT. Assigning certain groups of drives to each esoteric unit name used in the PARASORT JCL will in most cases ensure the required channel separation.

The procedure for creating PARASORT esoteric unit names consists of three steps:

1. Run the SYNCLIST report program to create a report of available tape drives and potential esoteric unit names.
2. Implement the esoteric unit names in the report.
3. Distribute the esoteric unit name table to developers of PARASORT applications.

The following three sections describe the steps in detail.

STEP 1: Examine the PARASORT Report Generated by SYNCLIST

The SYNCLIST report program executed as part of the normal SyncSort installation process generates a report that includes a PARASORT report to aid you in creating esoteric unit names for PARASORT.

If you don't have a copy of the SYNCLIST report on hand, or if you later need to regenerate the PARASORT report for a changed system configuration, simply run the SYNCLIST report program. The following shows sample JCL to execute SYNCLIST:

```
//STEP1      EXEC    PGM=SYNCLIST
//STEPLIB    DD      DSN=SYNCSORT.R11.SYNCLINK,DISP=SHR
//SYSPRINT   DD      SYSOUT=*
//SYSLIB     DD      DSN=SYNCSORT.R11.SYNCLINK,DISP=SHR
//           DD      DSN=SYNCSORT.R11.SYNCRENT,DISP=SHR
```

For more information on SYNCLIST, see "The SYNCLIST Report Program" on page 5.56.

The PARASORT report, which appears in the SYNCLIST report after the "Environmental / Control Block Information" section, provides information in two forms:

- The body of the report is for system programmers. It lists the configuration of the available tape drives along with suggestions for esoteric unit names for each device type and for each type of multiple input (2, 3 or 4 SORTPARn DD statements). The suggested esoteric unit names appear in quotes.
- The conclusion of the report is a table for distribution to PARASORT application programmers. The table displays the exact esoteric unit names that must be used in PARASORT applications.

The report is generated twice, under the headings "PART 1: ONLINE AND OFFLINE DRIVES" and "PART 2: ONLINE DRIVES ONLY."

The following shows a sample PARASORT report generated by SYNCLIST:

SYNCSORT PARASORT CONFIGURATION REPORT

CPU SERIAL 123456

01/18/2001

PART 1: ONLINE AND OFFLINE DRIVES

3420 TYPE DRIVES

AVAILABLE STRINGS AND ASSOCIATED CHANNEL PATHS

DRIVES 0120-012F, CHP=21

DRIVES 0220-022F, CHP=22

DRIVES 0320-032F, CHP=23

DRIVES 0420-042F, CHP=24

DRIVES 0520-052F, CHP=25

DRIVES 0620-062F, CHP=26

DRIVES 0720-072F, CHP=27

DRIVES 0820-082F, CHP=28

SUGGESTED ESOTERIC UNIT NAME ASSIGNMENTS

FOR 4-WAY INPUT:

"PAR241": 0120-012F,0220-022F

"PAR242": 0320-032F,0420-042F

"PAR243": 0520-052F,0620-062F

"PAR244": 0720-072F,0820-082F

FOR 3-WAY INPUT:

"PAR231": 0120-012F,0220-022F,0320-032F

"PAR232": 0420-042F,0520-052F,0620-062F

"PAR233": 0720-072F,0820-082F

FOR 2-WAY INPUT:

"PAR221": 0120-012F,0220-022F,0320-032F,0420-042F

"PAR222": 0520-052F,0620-062F,0720-072F,0820-082F

3480/3490 TYPE DRIVES

AVAILABLE STRINGS AND ASSOCIATED CHANNEL PATHS

DRIVES 0140-014F, CHP=4142

DRIVES 0240-024F, CHP=4344

SUGGESTED ESOTERIC UNIT NAME ASSIGNMENTS

FOR 4-WAY INPUT:

"PAR441": 0140-014F

"PAR442": 0140-014F

"PAR443": 0240-024F

```

    "PAR444": 0240-024F
FOR 3-WAY INPUT:
    "PAR431": 0140-014F,0240-024F
    "PAR432": 0140-014F
    "PAR433": 0240-024F
FOR 2-WAY INPUT:
    "PAR421": 0140-014F,0240-024F
    "PAR422": 0140-014F,0240-024F

3490E TYPE DRIVES

AVAILABLE STRINGS AND ASSOCIATED CHANNEL PATHS
DRIVES 01E0-01EF, CHP=E1E2E3E4

SUGGESTED ESOTERIC UNIT NAME ASSIGNMENTS
FOR 4-WAY INPUT:
    "PARE41": 01E0-01EF
    "PARE42": 01E0-01EF
    "PARE43": 01E0-01EF
    "PARE44": 01E0-01EF
FOR 3-WAY INPUT:
    "PARE31": 01E0-01EF
    "PARE32": 01E0-01EF
    "PARE33": 01E0-01EF
FOR 2-WAY INPUT:
    "PARE21": 01E0-01EF
    "PARE22": 01E0-01EF

3590 TYPE DRIVES

AVAILABLE STRINGS AND ASSOCIATED CHANNEL PATHS
DRIVES 0150-015F, CHP=0102

4-WAY PARASORT NOT POSSIBLE WITH 3590 TYPE DRIVES
3-WAY PARASORT NOT POSSIBLE WITH 3590 TYPE DRIVES
SUGGESTED ESOTERIC UNIT NAME ASSIGNMENTS
FOR 2-WAY INPUT:
    "PAR521": 0150-015F
    "PAR522": 0150-015F

```

SYNCSORT PARASORT ESOTERIC UNIT NAME TABLE				
CPU SERIAL 123456			01/18/2001	
PART 1: ONLINE AND OFFLINE DRIVES				

	3420	3480/90	3490E	3590
	-----	-----	-----	-----
4-WAY	PAR241	PAR441	PARE41	NOT
INPUT	PAR242	PAR442	PARE42	POSSIBLE
	PAR243	PAR443	PARE43	
	PAR244	PAR444	PARE44	
	-----	-----	-----	-----
3-WAY	PAR231	PAR431	PARE31	NOT
INPUT	PAR232	PAR432	PARE32	POSSIBLE
	PAR233	PAR433	PARE33	
	-----	-----	-----	-----
2-WAY	PAR221	PAR421	PARE21	PAR921
INPUT	PAR222	PAR422	PARE22	PAR922
	-----	-----	-----	-----
SYNCSORT PARASORT CONFIGURATION REPORT				
CPU SERIAL 123456			01/18/2001	
PART 2: ONLINE DRIVES ONLY				

.				
.				
.				

Figure 20. Sample PARASORT Report from SYNCLIST Output

Only the first few lines of Part 2 are shown in the sample report above. Part 2 is identical in format to Part 1. In general, Part 2 will list fewer drives than Part 1, since Part 2 lists only online drives, while Part 1 lists both online and offline drives.

The suggested esoteric unit names are within quotes (for example, "PAR241"). To comply with standards at your site, you may need to use esoteric unit names different from the suggested names.

The PARASORT esoteric unit name definitions ensure the required channel separation in most cases. You assign the indicated device strings to the names as described in STEP 2, below.

Even if all units within a device type already provide sufficient channel separation, it is recommended that you implement the PARASORT esoteric unit names and use them in your PARASORT applications. Doing so will allow you to respond easily to system configuration changes. If you use non-esoteric unit names in the JCL for your PARASORT applications (for example, "3480") and your tape configuration changes, the non-esoteric unit names may no longer provide sufficient channel separation. You would then need to change the JCL for all your PARASORT applications. On the other hand, if you use PARASORT esoteric unit names in all your PARASORT JCL, you don't have to modify that JCL. Simply change the esoteric unit name definitions. Do this by repeating the 3-step procedure for creating PARASORT esoteric unit names.

If there are insufficient channel paths available to run PARASORT using a particular tape device type, the report will indicate "NOT POSSIBLE" for that device. In certain cases, 4-way input may not be possible, but 3-way or 2-way input may be possible. For instance, there are only two available channel paths for 3590 devices in the sample report. This implies that only a PARASORT with 2-way input (SORTPAR1 DD and SORTPAR2 DD) may be used on those devices. Thus, only two esoteric unit names would need to be defined for 3590 drives.

If you want all tape drives to be available for PARASORT, then use Part 1 of the report, "ONLINE AND OFFLINE DRIVES." If you want to make only the online drives available for PARASORT, then use only Part 2 of the report, "ONLINE DRIVES ONLY."

Check that the drives listed in the part of the report you will use are acceptable to you for use by PARASORT. If they are acceptable, skip the following section, "Generating a Customized Report," and go to "STEP 2: Implement the Esoteric Unit Names" on page 7.8.

Generating a Customized Report

If you find that the report you want to use lists unacceptable drives, then you must generate a customized report that includes only the acceptable drives. However, you cannot simply drop the unacceptable drives from the esoteric unit name assignments. Doing so may invalidate the desired channel separation. Instead, use the following procedure to generate a new, customized report.

1. Rerun SYNCLIST, adding a new PARAOUT DD statement that defines a sequential data set or a partitioned data set member. A list of all tape drive unit numbers (one 10-byte record for each drive) will be written to PARAOUT, which may have any DSN. The following shows sample JCL for the PARAOUT DD statement:

```
//PARAOUT DD DSN=parasort.drives,DISP=(,CATLG),
//          UNIT=SYSDA,SPACE=(TRK,(1,1))

           or

//PARAOUT DD DSN=my_dsn(drives),DISP=OLD
//*          must be RECFM=FB,LRECL=10
```

2. Edit the PARAOUT DD data set or member to remove all drive numbers that are not eligible for PARASORT use. Simply delete the lines containing the drive numbers of ineligible drives. Do not make any other editing changes.
3. Change the SYNCLIST JCL so that it will read the PARAOUT DD. To do this, change the PARAOUT DD to PARAIN. The following example shows JCL for the PARAIN DD statement:

```
//PARAIN DD DSN=parasort.drives,DISP=SHR

           or

//PARAIN DD DSN=my_dsn(drives),DISP=SHR
```

The PARASORT report will be regenerated using only the drives listed in the PARAIN DD. The new report will contain only one part since you have specified all the acceptable drives.

4. Proceed to STEP 2, which implements the esoteric unit name definitions.

STEP 2: Implement the Esoteric Unit Names

The PARASORT esoteric unit name definitions may be implemented by using either the HCD process or an IOGEN. You will then be able to create PARASORT applications with the appropriate esoteric unit names.

STEP 3: Distribute the PARASORT Esoteric Unit Name Table to Programmers

A table, headed "SYNCSORT PARASORT ESOTERIC UNIT NAME TABLE," is printed at the end of each part (Part 1 and Part 2) of the PARASORT report. It is also printed at the end of any customized PARASORT report you generate.

The table should display the exact unit names application programmers must use on the SORTIN and SORTPARn DD statements for a PARASORT application. Once you have

implemented the suggested esoteric unit name definitions, distribute the appropriate table (from Part 1, Part 2, or customized report) to PARASORT application developers.

If you like, you can change the esoteric unit names from the suggested names before distributing the table.

Responding to System Configuration Changes

If the tape configuration at your installation changes, you may need to change the esoteric unit name definitions to ensure continued channel separation for PARASORT. To do this, simply repeat the 3-step procedure just described.

Chapter 8. Debugging

This chapter contains debugging information for both SyncSort for z/OS and for global Dynamic Storage Management (DSM).

Automatic Sort Retry Facility

Whenever possible, SyncSort will automatically retry a sort that fails because of a SyncSort internal error or certain system abends. The retry facility requires no JCL specifications or other intervention.

When an error occurs in an execution of SyncSort, the retry facility automatically sends messages to the console and to the unsuccessful sort's SYSOUT listing. These are messages WER426I and WER427I, and they say that an exception has occurred and that a recovery attempt is in progress. For more specific information, see the SyncSort message dialog (SS11MSG) or the message section of the *SyncSort for z/OS Programmer's Guide*.

SyncSort Termination Message

If any abend occurs while SyncSort's STAE or ESTAE is in effect, the following message will be received:

WER999A jobname,stepname,procstepname - UNSUCCESSFUL SORT xxxx y

xxxx is the abend code, and y can have the value "S" (system abend), "U" (user abend), or "W" (internally detected SyncSort error). The presence of the message does not necessarily

mean that SyncSort was responsible for the error. Since STAE or ESTAE enables SyncSort to gain control in the event of any abend, the error may have occurred in a user program or the operating system, or in SyncSort itself.

Dumps

If a SYSUDUMP or SYSABEND dump is produced, it should be interpreted as it would for any other program that issues a STAE or ESTAE. In particular, the ACTIVE RB chain is valid and represents the best source of the PSW and general registers at the time the error was detected.

For a sort that does not use tape SORTWK data sets, the SyncSort STAE/ESTAE routine can generate a SNAP dump with ID=255 for system abends and ID=nnn for internally detected errors, where "nnn" is the internal error number in decimal. This dump will be written to SYSUDUMP, SYSABEND, SPYSET, or the message data set (whose DDNAME is generally SYSOUT), depending on which is present.

Unlike in a system dump, the PSW AT ENTRY TO SNAP is meaningless for debugging purposes. Since SyncSort has detected an error and forced an abend, the registers may or may not be of use. If the termination was caused by a system abend, the PSW in the appropriate RB will be accurate as in any other system abend. However, if the termination was caused deliberately by SyncSort, the PSW will not be useful.

SyncSort deliberately terminated if the WER999A message ends with a "W" or if it contains a 1000-series user abend (for example, 1024 U).

The DEBUG Parm Facility

Certain SyncSort-detected errors simply produce an error message. In some of these cases, it may be useful to produce a dump for better diagnosis.

If RC16=ABE is set, a user abend 16 will be produced in case of an error, but no dump will be generated. In such cases, the diagnostic parameter DEBUG may be used to produce a SNAP dump (that is, a dump before any cleanup has taken place). DEBUG will also cause the sort to issue a user abend 16.

In the case of a system abend, a dump will be produced regardless of the parameters that are in effect. Should DEBUG be specified in this case, two dumps may be produced: a SNAP dump and the usual system dump. Both dumps generally have the same content.

In addition to a dump, DEBUG produces a listing of internal variables after each of the main phases, which may be useful to SyncSort for z/OS Product Services. The listing is produced on the SPYSET data set, if present. If only SYSOUT is present, the listing will not be produced, but the SNAP dump may be produced. To guarantee that both the listing and the dump will be produced, you must supply a SPYSET DD statement in the step. A SYSABEND statement will result in a full system dump, including a dump of the nucleus.

The DEBUG parm may be specified through any of the usual methods: EXEC parmlist, alternate parm facility (\$ORTPARM), PARMTABLE, or PARMEXIT.

Error Handling in Global DSM

SyncSort jobs will generally complete normally when a critical error or abend occurs in global DSM. The active SyncSort job will detect that global DSM is no longer available and will continue processing without attempting any further interaction with global DSM. However, if the DEBUG parameter is active on a SyncSort job step and an error or abend occurs in global DSM, the SyncSort job step will end with an internal abend (5DBW). This allows SyncSort to generate diagnostic information that will be helpful in debugging.

Abends in Global DSM

Two types of critical errors may occur in global DSM: a system abend or user-issued abend. In either case, global DSM will attempt to produce diagnostic information, and then will automatically terminate. When termination is complete, the message WER606I will be displayed on the console. At that point, global DSM will no longer be active, and all current and subsequent SyncSort jobs will automatically stop interacting with global DSM.

If a system abend occurs in global DSM, DSM will attempt to generate a SNAP dump to the SYSOUT data set as well as a standard SYSUDUMP. In certain instances, global DSM will also produce a diagnostic dump on the SYS1.DUMPxx data set.

If an error condition is detected internally in global DSM, DSM will attempt to generate a diagnostic dump on the SYSOUT data set and terminate with the USER abend code of 714 (hexadecimal).

Contacting Product Services

Syncsort Incorporated is proud of its reputation in the computer industry for customer service. You can reach us 24 hours a day, seven days a week if you have any questions about SyncSort for z/OS.

During general business hours (9 a.m. to 6 p.m. EST), a trained analyst is available directly. During evenings and weekends, an analyst will return your call in a timely manner for emergency situations. Our direct dial number for SyncSort for z/OS is 201-930-8260. Our e-mail address is zos_tech@syncsort.com

If you have a sort problem, try to get as much documentation as possible before calling, including listings and dumps. Often fixes can be given over the telephone and immediately zapped in. Documentation for the fix can then be mailed, e-mailed or faxed to you.

Appendix A. Messages

This appendix contains information about global DSM and license key messages.

Global DSM Message Format

Global DSM consists of a centralized administrative program that executes in its own address space and communicates with sorts in progress through the z/OS subsystem interface.

All messages issued by global DSM are routed to the console and have the form:

WERxxxxy message text

xxx is the message number and the y is either the letter A or the letter I.

The messages with A indicate a critical error. These messages are routed to the console with ROUTCDE=(2,11) and DESC=(2).

The messages with I provide information on global DSM execution or a condition not severe enough to warrant global DSM termination. These messages are routed to the console with ROUTCDE=(2,11) and DESC=(4).

Global DSM Messages

**WER600I SYNCSORT DSM SUBSYSTEM - UNSUPPORTED SYSTEM
COMMAND. REQUEST IGNORED**

EXPLANATION: Global DSM does not support the system command that was issued. The request has been ignored, and global DSM will continue normal operation.

WER605I SYNCSORT DSM SUBSYSTEM INITIALIZATION COMPLETE

EXPLANATION: Global DSM has been successfully started.

WER606I SYNCSORT DSM SUBSYSTEM TERMINATION COMPLETE

EXPLANATION: Global DSM has been terminated by a STOP command or an internally detected critical error. If a STOP command was not issued, refer to previous WERxxxxy messages for additional information.

WER607I SYNCSORT DSM SUBSYSTEM TERMINATION IN PROGRESS

EXPLANATION: Global DSM has received a STOP command and has initiated termination processing. WER606I will be issued when termination processing is complete.

**WER610A SYNCSORT DSM SUBSYSTEM TERMINATED DUE TO CRITICAL
ERROR**

EXPLANATION: One or more supporting functions in global DSM have ended with a critical error or abend. Global DSM will try to write a diagnostic dump to a SYS1.DUMPxx data set.

**WER612A SYNCSORT DSM SUBSYSTEM xxxxxxxx LOAD FAILURE,
CODE=yyy, RC=zzz**

EXPLANATION: The module xxxxxxxx cannot be successfully loaded. For an explanation of the codes yyy and zzz, see the appropriate IBM *Message Library: System Codes* manual for your system. Global DSM is terminated.

**WER614A SYNCSORT DSM SUBSYSTEM ALREADY ACTIVE. REQUEST
IGNORED**

EXPLANATION: Two global DSM subsystem interfaces with the same name may not be active on one system simultaneously.

WER615A	<p>SYNCSORT DSM SUBSYSTEM LIBRARY MUST BE APF AUTHORIZED</p> <p>EXPLANATION: The global DSM modules must reside in an APF authorized library.</p>
WER616A	<p>SYNCSORT DSM SUBSYSTEM - UNSUPPORTED OPERATING SYSTEM</p> <p>EXPLANATION: Global DSM requires a z/OS or OS/390 operating system.</p>
WER617A	<p>SYNCSORT DSM SUBSYSTEM - DSMHIST DD NOT DEFINED</p> <p>EXPLANATION: Global DSM requires a DSMHIST DD statement be defined in the SYNCDSM1 JCL procedure.</p>
WER618A	<p>SYNCSORT DSM SUBSYSTEM - HISTORY DATA SET UNINITIALIZED</p> <p>EXPLANATION: Global DSM's history data set must be initialized prior to the execution of the START command for the SYNCG100 JCL procedure. For more details, see “STEP 9: Initialize GLOBAL DSM's History Data Set” on page 3.7 if you are installing without SMP, or see “STEP 14: Initialize Global DSM's History Data Set” on page 4.11 if you are installing with SMP.</p>
WER619A	<p>SYNCSORT DSM SUBSYSTEM - INCORRECT HISTORY DATA SET FOR THIS SYSTEM ID</p> <p>EXPLANATION: A history data set in global DSM cannot be shared by different systems. You must allocate a separate history data set for each system, and activate it. For more details, see “STEP 8: Allocate Global DSM's History Data Set” on page 3.7 if you are installing without SMP and “STEP 13: Allocate Global DSM's History Data Set” on page 4.11 if you are installing with SMP.</p>
WER620A	<p>SYNCSORT DSM SUBSYSTEM - UNEQUAL MAINTENANCE LEVELS: xxxxxxxx, yyyy, zzzzz</p> <p>EXPLANATION: The load module xxxxxxxx of global DSM (GDSM) and the GDSM root module maintenance levels do not correspond. yyyy represents the maintenance level of the xxxxxxxx module; zzzzz represents the maintenance level of the root module.</p>

License Key Messages

The following are the messages directly related to the use of a SyncSort for z/OS license key.

**WER900A SYNCSORT 1.1 TPF_{xx} IS NOT LICENSED FOR SERIAL sssss,
TYPE mmmm mmm, MSU CAPACITY ccccc.**
or
**SYNCSORT 1.1 TPF_{xx} IS NOT LICENSED FOR SERIAL sssss,
TYPE mmmm, VERSION CODE vv.**

EXPLANATION: No valid license key for use on the specified machine was found, and the grace period for this error, noted by the WER903I warning message, has expired. A key must contain the correct information for both the serial number and the full model number. License keys are specified either in the KEY parameter of the SYNCMAC installation options macro, or included in a data set whose name is specified in the KEYDSN parameter of SYNCMAC.

ACTION: Execute the SYNCLIST program on the system where this message is occurring. Ensure that either the SYNCMAC KEY parameter or the data set named in the KEYDSN parameter has provided a valid key for this machine. If you require further assistance, contact SyncSort for z/OS Product Services with the SYNCLIST output available for reference.

WER901I **WARNING SYNCSORT 1.1 TPF_{xx} WILL EXPIRE IN nnn
DAYS**

EXPLANATION: The provided license key for this machine is only valid for the next nnn days. After that time, WER902A will be issued, and SyncSort will not execute.

ACTION: Contact the systems programmer in charge of SyncSort maintenance, or execute the SYNCLIST program on the system where this message is occurring and contact SyncSort for z/OS Product Services.

Note: For information on suppressing this message, see “Appendix C. Using a Data Set for SyncSort License Keys”. Also, see the KEYWARN and KEYMSG parameters of the SYNCMAC installation options macro.

WER902A SYNCSORT 1.1 TPF_{xx} HAS EXPIRED

EXPLANATION: The provided license key for this machine is no longer valid because the expiration date has passed. SyncSort will no longer execute.

ACTION: Contact the systems programmer in charge of SyncSort maintenance, or execute the SYNCLIST program on the system where this message is occurring and contact SyncSort for z/OS Product Services.

WER903I

**SYNCSORT 1.1 TPF_{xx} IS NOT LICENSED FOR SERIAL sssss,
TYPE mmmm mmm, MSU CAPACITY ccccc.**

or

**SYNCSORT 1.1 TPF_{xx} IS NOT LICENSED FOR SERIAL sssss,
TYPE mmmm, VERSION CODE vv.**

**SYNCSORT WILL STOP WORKING IN nnn DAYS UNLESS A
VALID KEY IS INSTALLED.**

EXPLANATION: No valid license key for use on the specified machine was found. License keys are specified in the KEY parameter of the SYNCMAC installation options macro, or included in a data set whose name is specified in the KEYDSN parameter of SYNCMAC.

SyncSort will allow sort processing to continue by issuing WER903I during a grace period after this error is first encountered. This will provide sufficient time to correct the problem by installing a valid key for this machine. If the grace period ends before a valid key is made available, WER900A will be issued and sort processing will terminate.

ACTION: Execute the SYNCLIST program on the system where this message is occurring. Ensure that either the SYNCMAC KEY parameter or the data set named in the KEYDSN parameter has provided a valid key for this machine. If you require further assistance, contact SyncSort for z/OS Product Services with the SYNCLIST output available for reference.

Note: For information on suppressing this message, see “Appendix C. Using a Data Set for SyncSort License Keys”. Also, see the KEYMSG parameter of the SYNCMAC installation options macro.

WER904I

**SYNCSORT 1.1 TPF_{xx} KEYUPDATE SUCCESSFUL;
xxxxxxxxxxxxxxxxxx SELECTED**

EXPLANATION: The KEYUPDATE parameter was specified, and SyncSort has successfully obtained a valid license key denoted by xxxxxxxxxxxxxxxxxxxx from SyncSort's key data set. The name of the data set was specified in the KEYDSN parameter of the SYNCMAC installation options macro.

WER905A**SYNCSORT 1.1 TPF_{xx} KEYUPDATE FAILURE: reason**

EXPLANATION: The KEYUPDATE parameter was specified, but SyncSort was unable to obtain a valid license key from SyncSort's key data set due to the specified reason. Possible reasons for this failure are:

1. The KEYDSN parameter of SYNCMAC was not specified when SyncSort was installed. KEYDSN, and not the KEY parameter, must be specified with the name of SyncSort's key data set when using the KEYUPDATE facility.
2. SyncSort was unable to dynamically allocate and/or read SyncSort's key data set. This can happen if you were editing the data set at the time of the KEYUPDATE run, or if the data set was not allocated as a fixed length 80-byte file.
3. No valid license key was found in SyncSort's key data set.
4. The SyncSort SVC was not available. SyncSort requires use of its SVC to perform the update.

ACTION: Ensure that the KEYDSN parameter has been correctly specified and that the data set is accessible and contains a valid license key. Also verify that the SyncSort SVC has been properly installed. If you require further assistance, execute the SYNCLIST program on the system where this message is occurring and contact SyncSort for z/OS Product Services with the SYNCLIST output available for reference.

WER906I**INVALID KEY DATA SET RECORD:
invalid record text**

EXPLANATION: One or more invalid records were found in the license key data set when performing KEYUPDATE. The first invalid record is displayed in the message text. Only comment statements, key statements and valid PARMs statements are permitted. All invalid statements are ignored.

ACTION: Correct any errors in the key data set record that was displayed in the message text and rerun the KEYUPDATE application.

WER907I**SYNCSORT EXPIRING LICENSE KEY WARNING MESSAGE
{ENABLED,DISABLED}****or****SYNCSORT INVALID LICENSE KEY WARNING MESSAGE
{ENABLED,DISABLED}**

EXPLANATION: These KEYUPDATE messages document whether SyncSort may issue certain license key warning messages. The default is for SyncSort to issue either the WER901I expiring license key warning message or the WER903I invalid license key warning message when applicable. During KEYUPDATE, a PARMS statement read from the key data set can disable the issuance of either of these messages. The WER907I message is intended to alert you that these warning messages may no longer be posted, though the warning period countdowns will continue. During the last seven days before the warning period ends, SyncSort will issue the warning messages **regardless** of whether they have been disabled. This is done to try to prevent termination of all SyncSort applications with either WER902A or WER900A.

ACTION: No action is required if both of these warning messages are enabled and you have a valid license key that is not expiring. If you do not have a valid key or if your key is expiring, contact SyncSort for z/OS Product Services as soon as possible to obtain a new license key and rerun the KEYUPDATE procedure using the new key. If any of the messages had been disabled, either remove the PARMS statement or set the warning message parameters to ON to re-enable the issuance of license key warning messages.

Appendix B. The SyncSort Installation Tape

The SyncSort installation tape is a standard-labeled tape with 40 files. The first file on the tape was placed on tape with IEBGENER. All the other files were placed using IEBCOPY.

If you need to modify your security profile to read the high-level qualifiers of the tape data sets, note the following:

- File 1 – High-level qualifier is SMPMCS
- Files 2 through 21 – High-level qualifier is SYNSRT
- Files 22 through 40 – High-level qualifier is SYNCSORT.

Appendix C. Using a Data Set for SyncSort License Keys

SyncSort license keys can be maintained either in the SYNCMAC installation options macro or in a separate data set. To maintain the keys in SYNCMAC, use the SYNCMAC KEY option. To maintain the keys in a separate data set, specify the name of the data set in the KEYDSN option of SYNCMAC. Using a data set eliminates the need to assemble and link modules each time a key is added or replaced, and avoids SMP/E involvement. The data set option is recommended for sites that have many license keys and experience frequent key changes.

The data set can also be used to tell SyncSort to suppress key-related warning messages. As a license nears expiration, or if there has been a change to a machine on which SyncSort is executed, WER901I and WER903I messages may be issued. If you want to suppress these messages, you can do so by using a PARMS statement in the key data set.

For more information on the SYNCMAC options, see “Chapter 5. Default Options”. The remainder of this appendix deals with the key data set.

Creating a Key Data Set

To build a data set for license keys, do the following:

1. Allocate and catalog a fixed-length, physical sequential data set of any block size with LRECL=80. Use the data set name you specified in the KEYDSN option of SYNCMAC. Note that every sort step needs to have READ access to this data set. WRITE authority should be limited to the systems programming staff that maintains SyncSort. SyncSort

dynamically allocates the data set, so there is no need for a DD statement in the sort JCL.

2. Specify each key in the data set in the first 16 bytes of each record, followed by an optional comment. Additional comments that begin with an asterisk (*) in column 1 are permitted. Keys must be typed in upper case.
3. Specify a PARMS statement if you want to control the issuance of key-related warning messages.

Disabling Messages

The PARMS control statement is used to disable or enable the issuance of WER901I expiring and WER903I unlicensed warning messages, except for the last 7 days of the warnings. For WER901I, these 7 days are the grace period *after* the documented expiration date.

PARMS Control Statement Format

The format of the PARMS control statement is illustrated below:

$\text{PARMS} \left[\text{EXPIRINGMSG} = \left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} \right] \left[, \text{UNLICENSEDMSG} = \left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\} \right] [, \text{SER} = \text{cccc}]$

Figure 21. PARMS Control Statement Format

The PARMS control statement applies to all machines that share the key data set, unless the SER parameter is used to specify each machine on which a message is to be disabled. The selected options apply to all releases of SyncSort for z/OS and all LPARs on the specified machines.

- All keywords and parameters must be in the first 71 columns of the statement and can begin in column 1 or have leading blanks (no statement labels are permitted).
- Continuation statements that end in a comma and continue on the next non-comment statement are permitted.
- All PARMS control statements must be placed before any records with keys in the data set. The statement will be ignored otherwise.

EXPIRINGMSG Parameter

The EXPIRINGMSG parameter controls the issuance of the WER901I message. The absence of the EXPIRINGMSG parameter or of the PARMS control statement in its entirety for a particular machine will cause the default (ON) to be put into effect.

When the WER901I message is written as a WTO, it will be written several times a day. The system will hold the message until it is deleted by the operator.

UNLICENSEDMSG Parameter

The UNLICENSEDMSG parameter controls the issuance of the WER903I message. The absence of the UNLICENSEDMSG parameter or of the PARMS control statement in its entirety for a particular machine will cause the default (ON) to be put into effect. Note that the third line of the SyncSort header, which contains license information, is not written when WER903I is issued, since there is no valid key. This header line will still be suppressed even if the WER903I message has been suppressed.

When the WER903I message is written as a WTO, it will be written several times a day. The system will hold the message until it is deleted by the operator.

SER Parameter (Optional)

The SER parameter specifies the particular machine on which the PARMS control statement settings will take effect. Only the last four characters of the machine serial number are required. If more digits are specified, only the four low-order digits will be used. If the SER parameter is not specified, the PARMS statement will apply to all machines.

Receiving an E-mailed File

You can request an e-mailed file from Syncsort containing your keys. This file will consist of records with a 16-character key followed by a comment that describes the encrypted license information. The file should be transferred to your mainframe system to be used as your key data set. An example of a key data set follows:

* SyncSort for z/OS Release 1.1 License Key Data Set for XYZ Corp.						
*						
*						
*Key	Release	Expiration	Serial	Machine	MSUs	
*		Date	Number			
*-----	-----	-----	-----	-----	----	
*						
F19BE9C138466B2C	1.1	28 Feb 2005	12345	IBM 2064 1C9	302	
418CB7CAD6E2441A	1.1	28 Feb 2005	6789A	IBM 2064 112	372	

Updating the Key Data Set

At initial installation time, SyncSort will automatically access the key data set if you have chosen the KEYDSN option. If you subsequently make a change to the data set, SyncSort will not become aware of this change immediately. SyncSort's normal practice is to access the data set once an hour. If you want to have your change take effect immediately, execute SyncSort with the following JCL on whichever LPARs or machines are affected by the change.

```
//KEYUPDT EXEC PGM=SYNCSORT, PARM='KEYUPDATE'  
//SYSOUT DD SYSOUT=*
```

Notes: If you are editing the key data set when SyncSort attempts to read it, SyncSort will not be able to dynamically allocate it and the KEYUPDATE procedure will fail.

Whenever the KEYUPDATE parameter is used, WER907I messages will be generated that indicate whether or not key-related warning messages have been disabled.

If you disable the messages, it is recommended that you enable them again as soon as you update the data set with your new key. Failure to do this could result in a very short warning period the next time you change a machine or renew a contract.

Index

\$

\$ORTPARM 5.4, 5.28, 5.51, 6.3, 8.3

A

ACCEPT 4.23

ACS 5.12

AD/CYCLE LE/370 3.2, 4.4

Alternate PARM Facility 8.3

ALTMSG 5.3, 5.50–5.51

ALTPARM 5.4, 5.51

ALTSEQ 5.4

ASR 5.57

B

BALANCE 5.9, 5.58

BetterGener 1.3, 3.3, 4.4, 6.1–6.3

 \$ORTPARM 6.3

 Activating 3.12–3.14, 4.17

 Error Messages 6.3

 Installing 3.12, 4.17–4.20

 Testing 3.12, 4.17–4.18

BKPTTIME 5.58

BLKSIZE Parameter 5.36

Block Size 5.35

BMSG 5.58

BRKPTIM 5.4

BSSIA11 4.4, 4.17–4.18

BSSIB11 4.4–4.5, 4.17–4.20

BSSIC11 4.5

BSSID11 4.5

BSSIL11 4.5

C

C 3.2

Century Window 5.5

CENTWIN 5.5

Changing Default Options 5.59–5.60

CLIST, Variable Length 2.2

CLISTS, SYNCINIT 2.2

CMP 5.7, 5.46

COB2LIB 3.2

COBEXIT 5.7

COBOL 3.2, 5.7

COBOL II Exits 4.5

COBOL/370 Exits 3.2

CPU 5.9, 5.58

D

DASD Space Requirements

 Non-SMP 3.3–3.4

- SMP/E 4.6–4.7
- Data Set for License Keys 5.15–5.16, C.1
- DCB Parameter 5.36
- DEBUG Parm Facility 8.2–8.3
- Debugging 8.1–8.3
- Default Options 5.59–5.60
 - See also Options
 - Changing for Non-SMP 5.59
 - Changing for SMP 5.60
- Disabling Key-Related Messages C.2
- DISP Parameter 5.36
- Distribution Tape
 - See Installation
- DSM 5.3, 5.8
 - See also Dynamic Storage Management (DSM)
- DSMWEND 5.3, 5.8
- DSORGDA 5.10
- Dumps 8.2–8.3
- DYNALOC 5.10
- Dynamic Storage Management (DSM) 1.4–1.6
 - Error Handling 8.3
 - Global DSM 1.5, 3.2, 4.3, 4.6, 5.8
 - Activating 3.8–3.11, 4.12, 4.14–4.16
 - Deactivating 3.11–3.12, 4.16
 - Functions 1.5–1.6
 - History Data Set 1.5, 3.4, 3.7–3.8, 3.10, 4.11–4.12, 4.15
 - Messages A.1–A.3
 - Modules 3.4
 - Local DSM 1.5–1.6, 5.8, 5.10
- DYNATAPE 5.12

E

- EBCDIC 5.4
- ELAP 5.9, 5.58
- ELPA 4.6
- ELPA Storage Requirements
 - Non-SMP 3.3
 - SMP/E 4.6
- EMSPACE 5.13, 5.57
- EQUALS 5.13
- Esoteric Unit Names 1.3–1.4, 3.3, 3.12, 3.15, 4.4, 4.16, 4.21, 7.2
- Esoteric Unit Names, Creating 7.1
- Execution Time Parameters 5.3, 5.56
- EXITNAME 5.14
- Exits, C 3.2
- Exits, COBOL II 4.5
- Exits, COBOL/370 3.2

- Exits, VS COBOL II 3.2, 4.5
- EXPIRINGMSG Parameter C.2

G

- GDSM, Initializing 3.7, 4.11
- GENERATE 4.17–4.18

H

- History Data Set 3.4, 3.7, 4.11

I

- IEALIMIT 5.31
- Installation
 - Procedure 3.4–3.9, 4.7–4.13
 - Tape 3.15, 4.21, B.1
 - Tape Sort 3.6
- Installation Options
 - See Options
- IO 5.9, 5.58
- IOERR 5.14–5.15, 5.51

J

- JCL, Sample 3.4–3.5, 4.8
- JCLCORE 5.52

K

- KEY 5.15
- Key
 - See License Key
- Key Data Set 5.15–5.16, C.1
- KEYDSN 5.16
- KEYMSG 5.16
- KEYUPDATE Parameter C.4
- KEYWARN 5.17

L

- LABEL Parameter 5.36
- Language Environment for MVS 5.18
- Libraries
 - DDnames for SYNCINIT 2.2
 - DDnames for Target and Distribution 4.5–4.6
 - Non-SMP 3.6

SMP/E 4.9
 SYNCINIT 2.2
 License Key
 Key Data Set 5.15–5.16, C.1
 KEY-Related Options 3.6, 4.11, 5.15–5.17
 Messages A.4
 SMP 4.2
 LISTJ 5.17, 5.50
 LISTT 5.17, 5.50
 LOCALE 3.2, 4.4–4.5, 5.17
 LRECL Parameter 5.36

M

Maintenance 3.13, 3.16, 4.18, 4.22–4.23
 MAXMERG 5.20
 MAXMERGE 5.59
 MAXSORT 5.4, 5.20–5.22, 5.25, 5.31, 5.33, 5.40,
 5.44, 5.58
 MAXWKSP 5.21
 MINCORE 5.2, 5.21, 5.57
 MINMERG 5.22
 MINMERGE 5.59
 MINWKSP 5.22
 MODSLNK 5.23
 MSG 5.23, 5.53
 MSGCLAS 5.24
 MSGDD 5.24, 5.50, 5.53
 MSGGENER 5.25
 MXROUTC 5.25

N

National Language 3.2, 4.4–4.5, 5.17–5.18
 Non-Resident 1.3, 3.3
 NOSNAP 5.25
 Notational Conventions 1.6
 NULLOFL 5.26
 NULLOUT 5.26

O

OLDGENER 4.20
 Optimization Modes 5.58
 Options
 SyncSort 3.4, 3.6, 4.10, 5.2–5.50, 5.59–5.60
 Tape Sort 4.11, 5.50–5.55, 5.59
 OUTSPCE 5.27

P

Parameters (DD Statements)
 BLKSIZE 5.36
 DCB 5.36
 DISP 5.36
 LABEL 5.36
 LRECL 5.36
 RECFM 5.36
 PARASORT 1.3, 3.12, 4.16, 7.1–7.9
 Creating Esoteric Unit Names 7.1–7.9
 Special Esoteric Unit Names 1.3
 SYNCLIST Report 7.2
 PARM Options
 SDB 5.35
 PARMEXIT 5.27, 8.3
 PARMNME 5.28
 PARMs 5.3, 5.27, 5.29, 5.56–5.62, 8.2
 PARMS Statement C.2
 PARMTABLE 8.3
 PARMTBLE 5.29
 PGMCORE 5.53
 PGMIEAL 5.31
 PGMRC16 5.31
 PREFIX 5.31
 PRINT121 5.32
 Product Services 8.3
 Program Product Options
 See Options

R

RC16 5.14, 5.31–5.32, 5.54
 RECFM Parameter 5.36
 RELEASE 5.33
 RELSMT 5.33
 RESET 5.33
 Resident 1.3, 3.3
 Retry, Automatic Sort 8.1
 RLSOUT 5.34
 ROUTCDE 5.34, 5.55, 5.57
 RSRVJI 5.3, 5.34
 RSRVTI 5.3, 5.35

S

Sample JCL 3.4–3.5, 4.8
 SCEELKED 3.2, 4.4
 SDB 5.35

SDBGENR 5.37
SECOND 5.37
SEDCBASE 4.19
SER Parameter C.3
SIBMBASE 4.19
SMF 5.38
SMFDSECT 3.15, 4.21
SMS 5.11
SOPAD 5.39
SOPADGN 5.39
SORTIME 5.4, 5.40
SORTIN Statement 5.36
SORTOFxx File 5.36
SORTOUT File 5.36
SORTTIME 5.59
SOTRN 5.40
SOTRNGN 5.41
Special Esoteric Unit Names 1.3–1.4, 3.3, 3.12,
3.15, 4.4, 4.16, 4.21, 7.2
Special Esoteric Unit Names, Creating 7.1
SS11ACCE 4.13
SS11ALLC 4.9
SS11ALOC 3.6
SS11ALOC. 3.15
SS11ALOE 4.7, 4.9
SS11APPE 4.10
SS11COBL 5.59
SS11GACY 4.17
SS11GAIN 4.17
SS11GAMV 4.18
SS11GATF 4.18
SS11GATI 4.17
SS11GBAC 4.20
SS11GBIN 4.18
SS11GBTF 4.19
SS11GBTI 4.18
SS11GBUF 4.19
SS11GBUI 4.18
SS11GENA 3.13
SS11GENF 3.13
SS11GENT 3.12
SS11GTSF 3.14
SS11GTSI 3.12
SS11HALO 3.6–3.7, 4.9, 4.11
SS11HINL 3.7, 4.11–4.12
SS11HPRO 3.8, 4.12
SS11INST 3.6
SS11INTE 4.9
SS11LOCL 5.60
SS11OPCH 5.59
SS11OPTE 4.11
SS11RECE 4.10
SS11SMP 4.10
SS11SVC 4.10
SS11SVEC 3.6, 3.15, 5.59
SS11TSTF 3.9, 4.13
SS11TVEC 3.6, 5.59
SS11TZAP 3.6, 5.59
START Command 3.8, 3.10–3.11, 4.12, 4.14, 4.16
STATOUT 5.42
STATREC 5.41, 5.58
STOP Command 3.10–3.11, 4.14, 4.16
STORCLAS 5.12
SUMOVFL 5.42
Support, Technical 8.3
Suppressing Key-Related Messages C.2
SVC 3.1, 4.3, 4.10, 5.2, 5.43, 5.50, 5.58
SYNCCPR 3.5, 4.8
SYNCG110 3.10
SYNCGENR 3.12–3.13, 4.17–4.19, 6.1
SYNCINIT 2.1–2.4
SYNCLIST 3.15, 4.21, 5.56, 7.2
PARASORT Report 7.2
SYNCMAC Macro 4.7, 5.1–5.3, 5.50, 5.57
SYNCVEC 4.21, 5.56
SYSMODs 4.5
SYSOUT 5.3, 5.24, 5.51, 5.53
SZBG111 4.18
SZBG112 4.19

T
Tape Density 5.55
Tape Sort 3.3–3.4, 4.4–4.6, 5.50–5.55, 5.57
TAPENME 5.44
Testing 3.7–3.9, 4.9, 4.11, 4.13
Installation Verification 3.7, 4.11
Libraries 3.6
Production Validation 3.8, 4.12
SVC 3.2, 4.3, 5.2
TIMESTAMP 5.32, 5.44
TITLE 5.44
TPF Maintenance Tapes 3.16, 4.22

U

UNINTDS 5.45
UNINTGN 5.45
UNLICENSEDMSG Parameter C.3
USERMOD 5.60
USERMODA 4.11
USERMODB 4.11

V

Variable Length CLIST 2.2
Visual SyncSort 1.6
VLTEST 5.46
VLTESTI 5.47
VS COBOL II Exits 3.2, 4.5
VSAMEMT 5.47
VSCORE 1.4, 5.2, 5.48, 5.58
VSCORET 5.2, 5.48, 5.58

W

WKSPCE 5.49

Y

Year Data 5.5

Z

ZDPRINT 5.50

We welcome comments on the usefulness and readability of this Manual. Your comments, along with suggested additions or deletions, will help us to improve future editions of the publications.

If you would like a reply, please indicate your name, business title, and business address. All comments and suggestions become the property of Syncsort Incorporated.

READERS' COMMENTS

SyncSort for z/OS
Installation Guide

SI-4302-3

Fold the form on the two lines, staple, and mail.
No postage stamp is necessary if form is mailed in the United States.



Fold

Fold



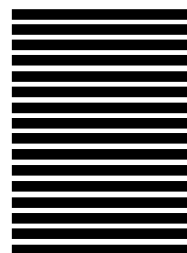
No postage
necessary
if mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 80 WOODCLIFF LAKE, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

Syncsort Incorporated
SyncSort for z/OS Product Services
50 Tice Boulevard
Woodcliff Lake, NJ 07677



Fold

Fold